TRABAJO FINAL DE CARRERA

TÍTULO DEL TFC: Monitorización de tráfico y encaminamiento en redes IP

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad Telemática

AUTOR: Alfonso Martín Román

DIRECTOR: David Rincón Rivera

FECHA: 15 de marzo de 2013

Título: Monitorización de tráfico y encaminamiento en redes IP

Autor: Alfonso Martín Román

Director: David Rincón Rivera

Fecha: 15 de marzo de 2013

Resumen

La monitorización y predicción del tráfico en redes de ordenadores es de vital importancia a la hora de evaluar el rendimiento y el servicio que ofrecen. Permite analizar el uso que se le está dando al sistema, obteniendo información del origen, destino y tipo de tráfico transmitido, así como si son suficientes los recursos de red disponibles. Con esto se consigue mejorar el funcionamiento de la red, detectar posibles anomalías y sus causas e implementar soluciones a la hora de optimizar los recursos disponibles.

El objetivo de este TFC es el estudio y despliegue de la herramienta de distribución libre `pmacct`. Ésta nos va a permitir evaluar y experimentar con las diferentes tecnologías que ofrecen para el estudio de monitorización de redes, como protocolos de encaminamiento (BGP), protocolos de recolección de datos (NetFlow) o sistemas de almacenamiento de la información (SQL). Se van a analizar las diferentes soluciones presentadas, determinando sus posibles limitaciones y deficiencias con el fin de comprobar si es una herramienta suficientemente robusta de cara a incorporarla a tareas docentes.

Para realizar el estudio con el máximo detalle, se presenta un esquema de red creado mediante un software de virtualización dónde se representará un entorno de red lo más realista posible en términos de, número de dispositivos, distribución de los mismos y protocolos desplegados tanto de encaminamiento como de monitorización.

Se asentarán las bases para futuros proyectos sobre esta herramienta, presentando diferentes ejemplos de configuración definiendo sus ventajas e inconvenientes, presentación de resultados obtenidos y descripción de los mismos.

Se ha conseguido determinar, que el software `pmacct` es una buena solución de cara a implementarlo en tareas docentes con el fin de estudiar técnicas relacionadas con la monitorización del tráfico.

Title: Traffic monitoring and routing in IP networks

Author: Alfonso Martín Román

Director: David Rincón Rivera

Date: March, 15th 2013

Overview

The monitoring and prediction of the traffic over computer networks is of vital importance in order to evaluate the performance and the service offered. This allows analyzing the usage of the system obtaining information about the origin, destination and type of transmitted traffic. With this, a better network performance can be achieved, by detecting possible faults and it causes and implementing solutions to optimize the available resources.

The goal of this degree thesis is the study and deployment of the free distribution tool pmacct. This tool allows us to evaluate and experiment with different technologies related to network monitoring, as well as routing protocols (BGP), data collection protocols (NetFlow) or data storing systems (SQL). This document describes and analyzes different solutions, defining its pointing out their limitations and deficiencies in order to test if it is a sufficient-robust tool with the purpose of using it in teaching tasks.

In order to perform the study with the maximum detail lever, we present a network testbed created with virtualization software. The scenario is as realistic as possible on terms of number of devices, their distribution and the protocols used as much for routing and for monitoring.

This thesis can be the basis of new studies about the aforementioned tools, showing up different examples of configuration, defining their advantages and disadvantages, showing the obtained results and their description.

We conclude that pmacct is a good solution to deploy and implement teaching activities in order to study techniques related with traffic and routing monitoring.

ÍNDICE

INTRODUCCIÓN

Estudiar el funcionamiento de la red sobre la que estamos trabajando es de vital importancia a la hora de optimizar sus recursos. De esta materia se encarga la gestión de red. Ésta analiza y controla el uso que se le está dando a los servicios proporcionados en una red con los objetivos entre otros de detectar errores, corregirlos en el menor tiempo posible y anticiparse a futuros problemas de congestión por el aumento de uso de recursos y así poder planificar nuevas estrategias de redimensionado que sean lo más transparentes posibles de cara al usuario.

Una de las partes de la gestión de red es la monitorización del tráfico. Ésta se encarga de analizar y observar el estado de los recursos gestionados por la red. Para ello es indispensable realizar medidas sobre el tráfico que está circulando en la red, obteniendo información como el origen, destino, cantidad y tiempo de transmisión del mismo.

Los operadores de red suelen basar sus operaciones de gestión en analizar la **carga de los enlaces**, es decir, medir la cantidad total de tráfico que está circulando por un par de nodos directamente conectados en un tiempo determinado. Es una primera medida de gestión relevante pero no suficiente, ya que no estamos controlando parámetros como el origen, el destino o el tipo de tráfico que se transmite. La carga de los enlaces se suele representar en forma de *weathermaps*, como el que presentamos a continuación, correspondiente a RedIRIS (Red española de interconexión de recursos informáticos) [1].

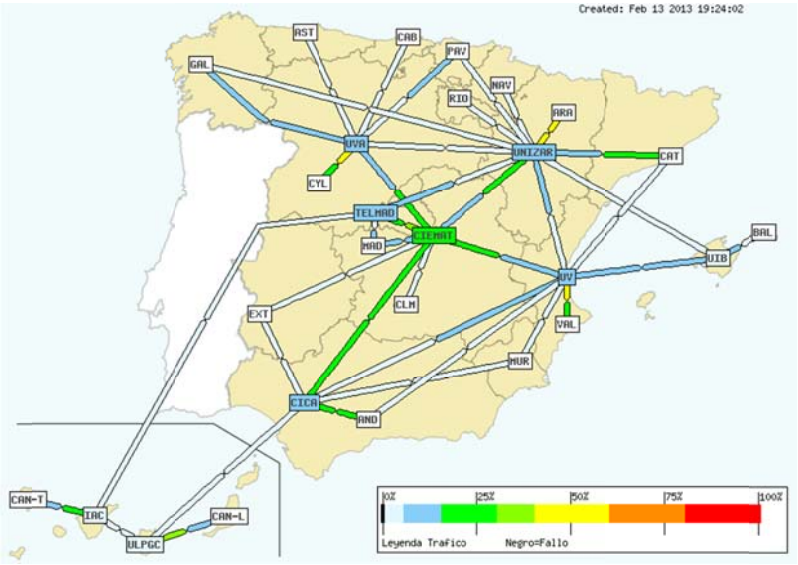


Fig. 1 Carga de los enlaces de RedIRIS (porcentaje de uso en intervalos de 5 minutos) para el día 13/02/2013 a las 19:24

Las matrices de tráfico ofrecen mucha más información. Éstas describen tres parámetros fundamentales en las operaciones de gestión: origen y destino del

tráfico y volumen de tráfico generado en un tiempo determinado. A diferencia del *weathermap,* con las matrices de tráfico obtenemos información del intercambio de tráfico entre todos los pares de nodos origen-destino. Dada una red con N nodos, la matriz tiene dimensiones NxN= N^2, correspondientes a todos los pares origen-destino. En la Fig.2 se presenta un ejemplo correspondiente a la red norteamericana Abilene [2], obteniendo la carga total del tráfico en un espacio determinado de tiempo, en este caso 5 minutos, entre todos los pares de nodos.

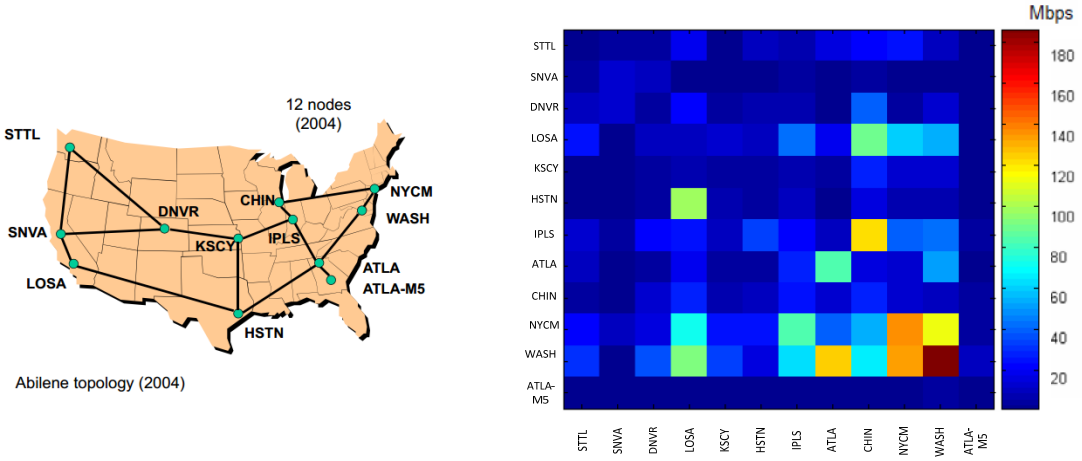


Fig.2 Matriz de tráfico de Abilene en 01/03/2004, de 00:00 a 00:05 (5 minutos)

Las matrices de tráfico ofrecen una información mucho más completa que la mostrada por la carga de los enlaces, y son una información vital de cara al diseño y optimización de la red. Por ejemplo, dado un encaminamiento permiten dimensionar los enlaces; o bien se puede calcular el encaminamiento óptimo para balancear la carga de la red. La carga de los enlaces, en cambio, ofrecen mucha menos información, ya que es dependiente del encaminamiento concreto que se está dando en ese momento en la red.

La relación entre topología, encaminamiento y matriz de tráfico está definida por la siguiente ecuación $Y = Ax$, dónde:

- Y es el vector de cargas de enlace. Su longitud es *K*, donde *K* representa el número de enlaces en la red. Nótese que los enlaces son unidireccionales, y que entre dos routers siempre habrá dos enlaces independientes. Por ejemplo, en la Fig.1, entre Barcelona y Zaragoza hay dos enlaces. Uno para cada sentido, con cargas diferentes.
- A es la matriz de encaminamiento. Puede tomar valores binarios {0,1}. Se define A_{ij}=1, si el enlace *i* pertenece a la ruta asociada al par *j*. Tiene unas dimensiones de K x N^2, definiendo *N* como el número nodos.
- X es la matriz de tráfico, expresada como un vector de longitud N^2, dónde x_{ij} define el tráfico asociado al par *i-j* origen – destino.

Herramientas disponibles: software y protocolos

Para realizar este tipo de medidas se han desarrollado diferentes protocolos y software. Se van a mencionar los dos protocolos más utilizados, que son:

a) Protocolo SNMP (*Simple Network Management Protocol*) [5], que nos va a permitir realizar medidas de carga de enlace y que va a poder ser gestionado gráficamente gracias a un software como Cacti.

b) Por otro lado tenemos el protocolo NetFlow [6]. Monitoriza flujos IP extremo a extremo, y ofrece mucha más información que SNMP, incluyendo valores como IP origen e IP destino, puertos, flags, etc. Nos va a permitir realizar medidas como las ya presentadas matrices de tráfico y que se podrán trabajar a través del software `pmacct`.

Objetivos del TFC

Los objetivos marcados en este TFC han sido el estudio y despliegue de la herramienta `pmacct` [7]. Nos va a permitir estudiar diversas tecnologías relacionadas con la monitorización de tráfico como pueden ser los protocolos de encaminamiento BGP y OSPF o protocolos de gestión de tráfico como NetFlow y SNMP. Se pretende hacer un análisis en profundidad de la herramienta, probando sus diferentes posibilidades y analizando sus resultados con el objetivo de cerciorarnos de que es una herramienta robusta y completa de cara a poder incorporarla en tareas docentes.

El estudio de la herramienta se ha llevado a cabo sobre un escenario creado mediante un software de virtualización. Se han creado diversas máquinas virtuales interconectadas mediante protocolos de encaminamiento interno y externo para focalizarlo lo más posible a la realidad. Se han utilizado tanto herramientas presentadas por el software `pmacct` como herramientas independientes ya presentes en la red con el fin de describir sus diferencias y corroborar la robustez de las primeras.

Se ha conseguido determinar que el software `pmacct` es una buena solución para incorporarse a tareas docentes, ya que los resultados obtenidos tanto de protocolos de encaminamiento, protocolos de gestión y software de almacenamiento de información han sido satisfactorios. Presenta una serie de deficiencias que han sido en gran medida resueltas gracias a soluciones propuestas dentro del mismo software con el paso del tiempo.

El resto del presente documento se ha organizado de la siguiente manera: en un primer capítulo se ha profundiza sobre teoría de gestión de redes y en concreto sobre monitorización de tráfico, describiendo los dos protocolos más utilizados en dicha tarea, como son NetFlow y SNMP. En el segundo capítulo se presenta el escenario a desarrollar y él software utilizado para poder llevarlo a cabo, incluyendo tanto software de virtualización como de encaminamiento y de gestión de redes. En el tercer capítulo se describen los archivos de configuración que se han generado y cuál es el funcionamiento de los mismos y para finalizar se presentan la serie de pruebas que se han llevado a cabo

para comprobar el funcionamiento de `pmacct` y analizar sus resultados. La memoria finaliza con las conclusiones y las líneas futuras de desarrollo. También se incluyen anexos con información sobre la instalación del software utilizados y los archivos de configuración creados en los mismos.

CAPÍTULO 1. MONITORIZACIÓN DE TRÁFICO

En este capítulo vamos a presentar el concepto de la gestión de redes, de qué se encarga y cuáles son sus objetivos, presentando especial énfasis en la parte de monitorización de tráfico. Se van a presentar las dos técnicas más utilizadas para el estudio y análisis de la monitorización de tráfico y se van a describir sus principales diferencias. Para finalizar se presentará una de las utilidades que tienen estas aplicaciones, que son la generación de matrices de tráfico.

1.1. Gestión de redes

1.1.1 Definición y utilidad

La gestión de redes se define como el conjunto de tareas que se van a encargar de controlar y vigilar los recursos de telecomunicación presentados en una red, con el objetivo final de garantizar un nivel de servicio óptimo al menor coste posible [8].

Sus objetivos son:

- Optimizar la operación de la red con mecanismos de control y monitorización, resolución de problemas y suministro de recursos.

- Realizar un uso eficiente de la red y de sus recursos como puede ser saber el porcentaje de utilización de la conexión al proveedor de servicios de internet.

- Configurar una red segura, por ejemplo, evitar el acceso a la información por personas ajenas, registrar intentos de acceso no autorizados.

- Controlar y actualizar cambios en la red como pueden ser cambios de topología, y que sean lo más transparentes posibles de cara al usuario, detectar que un router o switch no funciona como debe.

Las herramientas de gestión de red se basan en el paradigma “Gestor – Agente”. Los componentes de una red se pueden clasificar en dos grandes grupos:

- Gestores → Son los elementos que interaccionan con los operadores y llevan a cabo las acciones necesarias para cumplir con las tareas por ellos invocadas.

- Agentes → Son los encargadas de realizar las tareas de gestión solicitadas por los gestores de red.

El principio de funcionamiento de los sistemas de gestión de red se basa en el

intercambio de información entre los nodos gestionados definidos como agentes y los nodos gestores. Los agentes recogen información del estado y características de funcionamiento de un determinado recurso de red en los nodos gestionados. Mediante un protocolo de gestión de red el gestor ordenará al agente que realice una serie de operaciones con las que podrá conocer el estado de los recursos y con ello poder realizar una serie de operaciones que influirán en el comportamiento del equipo.

Fig. 1.1 Esquema paradigma gestor - agente

1.2. Monitorización

La monitorización es la parte de la gestión de red que se encarga de analizar y observar el estado de los recursos gestionados. Se definen cuatro fases:

- Definir la información de gestión que se va a monitorizar.
- Acceder a la información de monitorización. Las aplicaciones de monitorización van a utilizar los servicios ofrecidos por un gestor para acceder a los datos monitorizados mantenidos por un agente. La comunicación se va a llevar a cabo mediante los protocolos de gestión.
- Definir la política de monitorización. Se podrá distinguir en dos casos:
    - Sondeo → El gestor realizará consultas periódicas al agente

sobre los datos de monitorización.

- Informe basado en eventos → Los agentes, informarán a los gestores de manera autónoma cuando se produzca un cambio de estado significativo.

- Procesado de la información → Es la última etapa y más importante. Con ella se buscan los siguientes objetivos:
  - Detección de fallos y la corrección de los mismos en el menor tiempo posible, como puede ser, la caída de un enlace.
  - Optimización del uso de la red. Un ejemplo podría ser, optimizar el uso de ancho de banda o el balanceo de carga entre los nodos y enlaces de la red.
  - Gestionar de manera óptima los componentes del sistema.
  - Planificar posibles crecimientos del sistema.

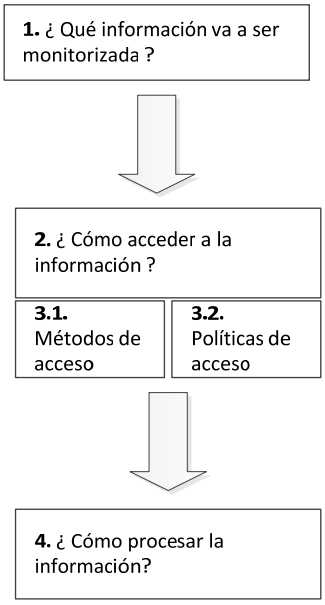


Fig. 1.2 Fases en la monitorización de red

A continuación se van a presentar dos de las técnicas más utilizadas para la obtención de información en tiempo real en redes IP: NetFlow y SNMP.

1.2.1. NetFlow

NetFlow [2, 9, 10] es el nombre que se da tanto a una aplicación como a un protocolo de red desarrollado por Cisco Systems que permite la recolección de tráfico de red.

La tecnología NetFlow describe la manera en que un router o switch exporta

estadísticas sobre el tráfico que circula por sus interfaces de red, mediante la generación de registros Netflow (denominados flujos) que se transportan vía datagramas UDP (User Datagram Protocol) o SCTP (Stream Control Transmission Protocol) a una máquina recolectora. La capacidad de exportar flujos no está disponible en la totalidad de routers Cisco. Otros fabricantes como Juniper o Ericsson ofrecen productos similares como Jflow/cflow [11] o Rflow [12]

Un flujo (*flow*) de datos, se define como un conjunto de paquetes IP que combinan una misma serie de atributos en un intervalo de tiempo. Los atributos más representativos son los siguientes (Figura 1. 3):

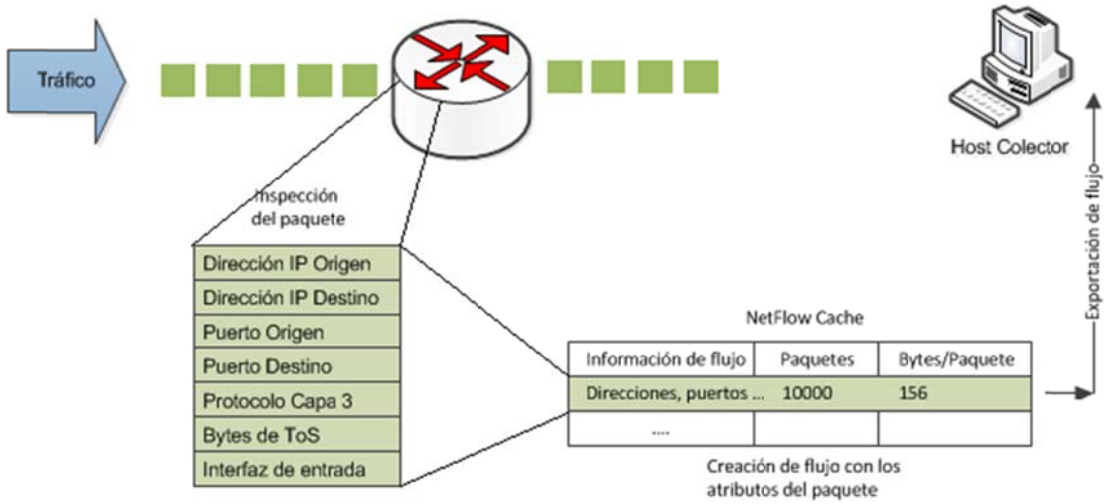


Fig. 1.3 Registro NetFlow [9]

- Dirección IP origen → Nos indica quién ha generado el tráfico
- Dirección IP destino → Quién es el destinatario de dicho tráfico.
- Puerto origen
- Puerto destino

  } Marca la aplicación que utiliza el tráfico.

- Tipo de protocolo en la capa de red.
- Bytes ToS (*Type of Service*). → Nos indica la prioridad del tráfico.
- Interfaz de entrada al router.
- Paquetes y bytes → Nos indica la cantidad de tráfico en el flujo.

Estos son los atributos para un registro NetFlow v5. Otras versiones NetFlow pueden incorporar información adicional como NetFlow v7 que incorpora un campo de identificador de router origen, o NetFlow v9 que incorpora un campo

de identificación de siguiente salto BGP, entre otros y permite definiciones más flexibles de flujo, basadas en plantillas.

1.2.1.1 Arquitectura

Se distinguen tres componentes básicos en la arquitectura de análisis/monitorización basadas en NetFlow:

- Exportador:

  Router o switch con la capacidad de generar registros NetFlow. Prepara la información para ser exportada al colector vía UDP.

- Colector:

  Dispositivo que escucha en un puerto UDP determinado y que tiene la capacidad de agregar información de acuerdo con los criterios establecidos por el operador de red.

- Analizador:

  Dispositivo encargado de analizar, mostrar y filtrar los flujos recibidos por el colector.

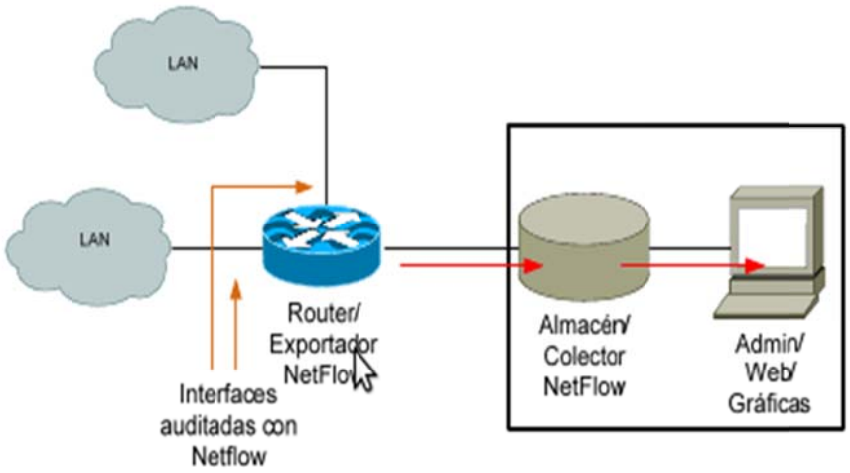


Fig. 1.4 Arquitectura NetFlow [9]

1.2.1.2 Formato de los registros

Existen diferentes versiones de registros NetFlow, resumidos en la Fig. 1.5. En nuestro estudio se ha optado por utilizar la versión 5, ya que es la más extendida.

Versión	Información
v1	Versión original ya obsoleta limitada a IPv4 y sin entender máscaras de red
v5	Versión más extendida y utilizada en números routers, solo entiende IPv4.
v7	Igual que v5 añadiendo un campo de identificación de router origen.
v8	Agregación de flujos sólo para información presente en v5
v9	Última versión NetFlow. Basada en plantillas. Orientada a flujos IPv6 y MPLS

Fig. 1.5 Tabla de versiones NetFlow [9]

A continuación se presenta un registro NetFlow V5 donde aparecen todos los campos clave (Figura 1.6).

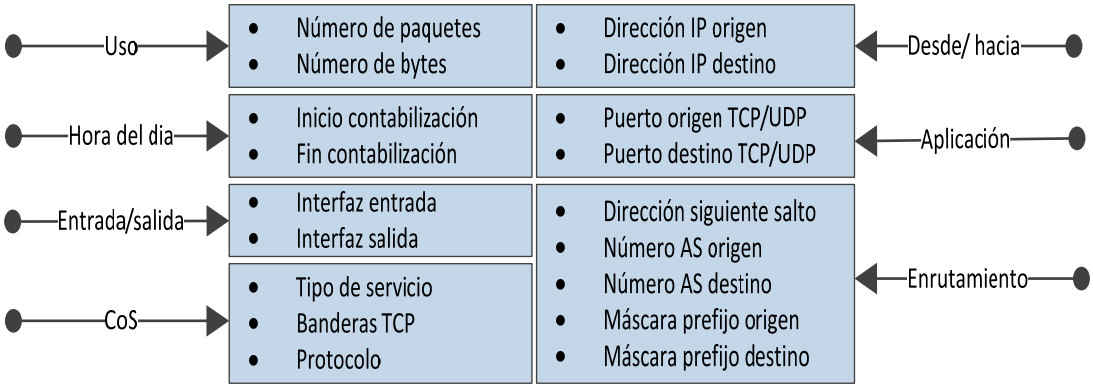


Fig. 1.6 Datagrama de exportación NetFlow V5 [10]

Se presenta la obtención de un contador total de paquetes (número de paquetes) y bytes (número de bytes) en un tiempo definido de inicio de flujo (Inicio contabilización) y último paquete recibido (Fin contabilización) transmitido por el usuario (Dirección IP origen) hacía el destinatario (Dirección IP destino) utilizando los puertos (Puerto origen TCP/UDP – Puerto destino TCP/UDP).

Respecto al encaminamiento se presenta la obtención de información como puede ser, cual es el siguiente salto del paquete (Dirección IP siguiente salto) con origen en el sistema autónomo (Número AS origen) y destino (Número AS destino).

La Fig. 1.7 presenta una captura real, tomada en nuestro sistema, donde se puede ver cuando fue generado el flujo (timestamp), la dirección IP origen (srcaddr), dirección IP destino (dstaddr), el número de paquetes (packets) y bytes del flujo (octets) entre otras.

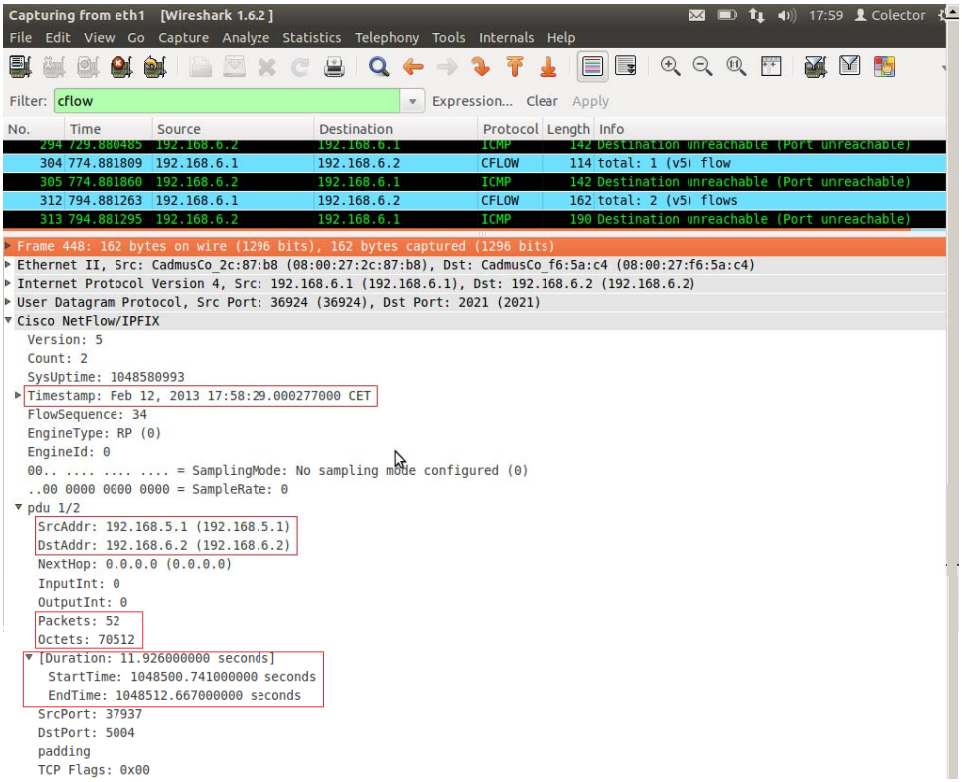


Fig. 1.7 Captura Wireshark

En la figura 1.7 también podemos verificar el tiempo de duración del flujo y el fin del mismo. Se entiende que un flujo ha finalizado cuando en un determinado tiempo, típicamente 15 segundos, no se ha observado tráfico. Otras maneras de determinar el fin de flujo es cuando se encuentra activo durante un tiempo superior a 30 minutos, se recibe una bandera de fin de conexión TCP o cuando el dispositivo que procesa la información (*router/switch)* no es capaz de recibir más flujos por problemas de congestión. Todos estos valores son configurables en el dispositivo agente donde se está recogiendo la información. La Fig. 1.8 muestra los valores relacionados con la detección de flujos que se han adoptado en nuestro sistema.

```
INFO ( default/nfprobe ):            TCP timeout: 3600s
INFO ( default/nfprobe ):  TCP post-RST timeout: 120s
INFO ( default/nfprobe ):  TCP post-FIN timeout: 300s
INFO ( default/nfprobe ):            UDP timeout: 300s
INFO ( default/nfprobe ):           ICMP timeout: 300s
INFO ( default/nfprobe ):        General timeout: 3600s
INFO ( default/nfprobe ):       Maximum lifetime: 604800s
INFO ( default/nfprobe ):         Expiry interval: 60s
```

Fig. 1.8 Valores adoptados para determinar el fin de un flujo

Se define fin de flujo en una conexión TCP cuando no se recibe ningún paquete en 1 hora (TCP timeout) o 2 minutos en el caso de recibir una bandera de reset (TCP post-RST timeout) o 3 minutos en el caso de bandera de fin (TCP post-FIN timeout). El tiempo de vida máximo de un flujo de 7 dias (Maximum lifetime). En los protocolos UDP e ICMP se definen valores de fin de flujo menores a los de TCP (UDP timeout/ ICMP timeout), ya que son protocolos no

orientados a conexión, es decir, el envío de datos se realiza sin previamente haber establecido una conexión. A diferencia del protocolo TCP, no existen parámetros que indiquen el fin de envío de paquetes como puede ser la bandera FIN en TCP lo cual hace complicado detectar un fin de flujo. Por ello se entiende que un valor igual a `TCP post-FIN timeout` es suficiente para determinar que el envío de paquetes ha finalizado. Para finalizar se especifica al agente el tiempo que debe esperar para transmitir los datos al colector (Expiry interval).

1.2.1.3 Principales beneficios de Netflow

NetFlow aporta [10]:

- Monitorización de la Red:

  Debido al análisis de flujos en cada nodo gestionado, va a permitir relacionar patrones de tráfico, con lo que el gestor de red va a poder anticiparse a posibles deficiencias del sistema como puede ser la falta de ancho de banda debido al incremento del consumo en los recursos proporcionados.

- Monitorización de aplicaciones:

  Con información como la de “puerto origen” o “puerto destino” va a permitir al gestor de red obtener información del consumo de recursos por parte de las aplicaciones utilizadas en el sistema. Con esto va a poder rediseñar los servicios ofrecidos e incorporar los recursos más demandados como pueden ser los asignados a un servidor de correo, o a un servidor web, entre otros.

- Monitorización de usuarios:

  Con información “IP origen” e “IP destino” va a permitir al gestor obtener información de qué usuarios están utilizando los recursos y como los están utilizando. Va a permitir rediseñar estrategias de acceso como puede ser bloquear la utilización de un servicio de la red o detectar problemas de seguridad entre otras.

- Planificación de la red:

  Con información como “interfaz de entrada” o “interfaz de salidad” va a permitir al gestor anticiparse a posibles crecimientos de la red, ya sea en dispositivos como en puertos y ancho de banda.

- Análisis de seguridad:

  Permite detectar anomalías en el tráfico de la red.

- Contabilidad y facturación:

Debido a sus detalladas estadísticas permite al proveedor de servicios facturar en función de tiempo y modo de uso de la aplicación.

- Almacenamiento de datos NetFlow:

  Para futuros análisis de los usos que se le han dado a los servicios ofrecidos y con ello realizar mejoras sobre los mismos.

1.2.1.4 Principales inconvenientes de NetFlow

El principal inconveniente que presenta NetFlow es el consumo de CPU en el dispositivo que opera a la hora de procesar la información recibida en los registros NetFlow. Esto se agrava en el caso, de que un nodo reciba una gran cantidad de flujos, como puede ser un router que opere en una red WAN y pierda eficiencia en tareas de encaminamiento por la realización de procesar la información recibida. NetFlow presenta una solución, Sampled Netflow, que basa su funcionamiento en procesar una fracción de los paquetes recibidos. La frecuencia de muestreo puede variar entre 1/65535 y 1, es decir, muestrear un rango desde sólo 1 de cada 65535 paquetes recibidos hasta la totalidad de éllos. En el caso del muestreo, se sufrirá la correspondiente pérdida de precisión en la exactitud de las medidas:

a) No detección de flujos porque no se ha muestreado ningún paquete correspondiente a ellos.

b) Estimación incorrecta del número de paquetes y bytes correspondientes a un flujo detectado.

1.2.2. SNMP (Simple Network Management Protocol)

SNMP [13] forma parte de la familia de protocolos TCP/IP. Facilita el intercambio de información de administración entre dispositivos de red y permite supervisar el funcionamiento de los equipos con el fin de diagnosticar y resolver posibles problemas.

Utiliza el protocolo de transporte UDP para enviar mensajes entre administradores y agentes.

1.2.2.1 Componentes y principios de funcionamiento

Una red administrada a través de SNMP está compuesta por tres componentes fundamentales:

- Nodos gestionados o elementos de red:

  Puede ser cualquier sistema que tenga algún tipo de conexión de red

como un router, impresora, un terminal, etc. Recogen la información de administración que solicita el agente y se envía a un NMS (*Network Management System*, definido más adelante).

- Agente:

  Aplicación de gestión de red ubicada en un dispositivo de red monitorizado que se encargará de recopilar la información de los eventos y comunicarse con el gestor.

- Sistema de administrador de red (*Network Management System,* NMS):

  Terminal que es capaz de enviar peticiones SNMP y recibir y procesar respuestas SNMP.

Los dispositivos administrados son supervisados y controlados usando cuatro comandos SNMP básicos:

- Comando de lectura:

  Permite supervisar los elementos de la red examinando las diferentes variables que son mantenidas por los dispositivos administrados.

- Comando de escritura:

  Permite controlar los elementos de la red y modificar los valores de las variables almacenadas dentro de los dispositivos administrados.

- Comando de notificación:

  Permite a los dispositivos administrados reportar los eventos de forma asíncrona. Cuando un evento tiene lugar, un dispositivo administrado envía una notificación al sistema de administrador de red.

- Operaciones transversales:

  Se utilizan para determinar que variables son soportadas por un dispositivo administrado y recoger la información en tablas de variables.

1.2.2.2 Arquitectura

Una tabla de variables o MIB (*Management Information Base)* es una base de datos que va a permitir tener acceso a la información de gestión guardada en el dispositivo. Es una base de datos con estructura en árbol, adecuada para gestionar diversos grupos de objetos. SNMP proporciona un mecanismo para acceder a los objetos de la MIB de manera que puedan ser consultados y modificados.

La tabla MIB se construye a partir de un elemento básico que es el identificador

de objeto o OID. Tiene una estructura similar a la de DNS, una secuencia de números separada por puntos. SNMP v2 tendrá el identificador 1.3.6.1.6 (Figura.1.10).

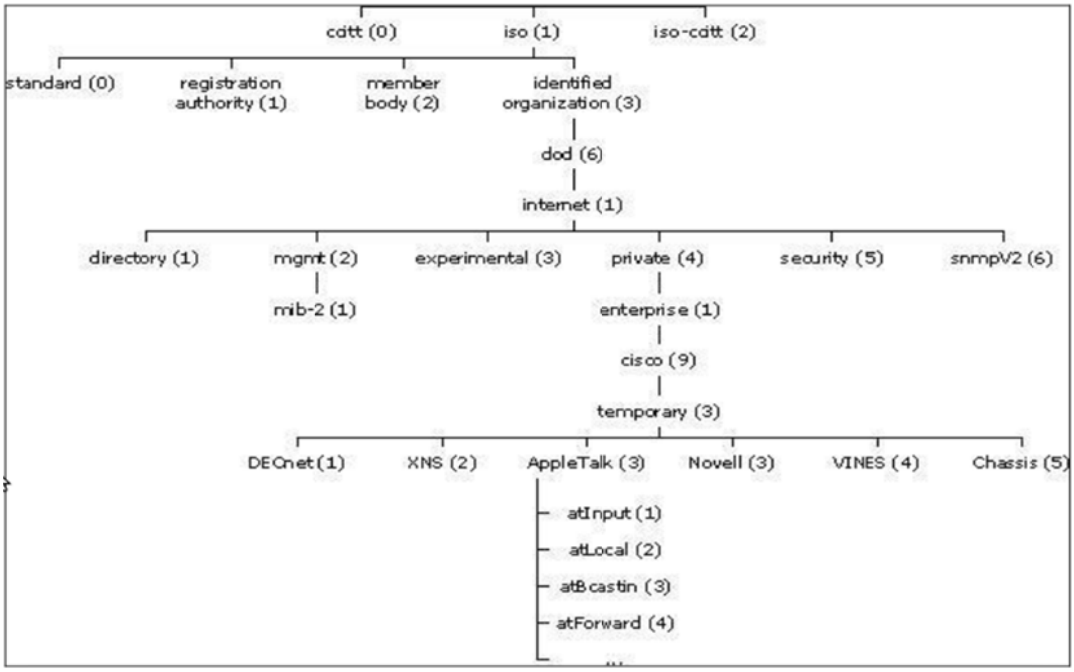


Fig. 1.9 Árbol MIB [7]

1.2.2.3 Registros SNMP

En el protocolo SNMP se han definido cinco tipos de mensajes entre gestor y agente:

- Get request → El gestor pregunta al agente sobre una lista de variables que desea leer de una MIB.
- Get Next request → Indica una comando de cómo leer secuencialmente una MIB.
- Set request → Se utiliza para notificar el cambio de una o más variables.
- Get response → Mensaje utilizado por el agente para responder a un mensaje de Get request, Get Next request o Set request.
- Trap → Mensaje utilizado para notificar que ha ocurrido un cambio significativo como, por ejemplo, la caída de un enlace.

A continuación presentamos una captura real de nuestro escenario utilizando el software Cacti para representar gráficamente los datos de los reportes SNMP correspondientes a la carga de un enlace (Figura 1.11)

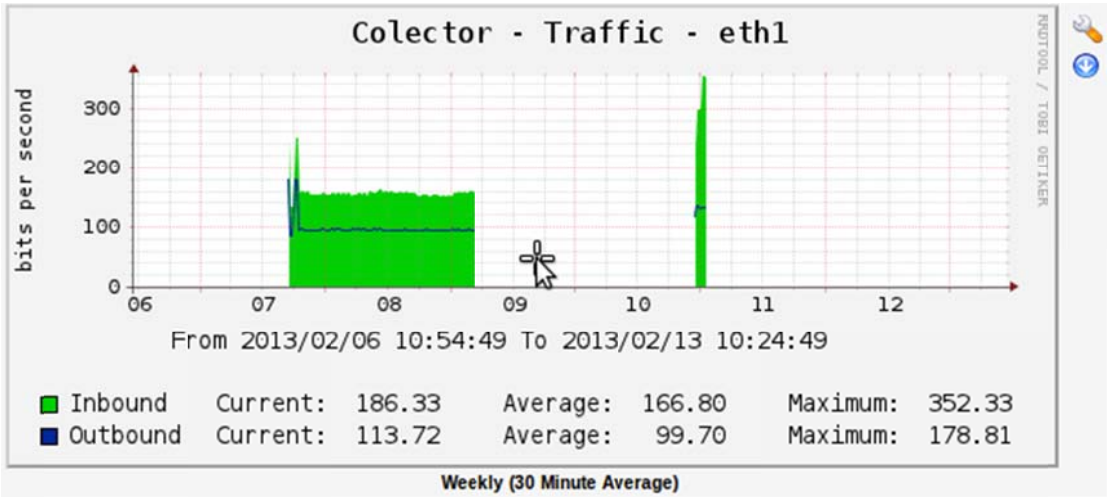


Fig.1.10 Captura de pantalla de Cacti.

1.2.2.4 Diferencias respecto a NetFlow

La principal diferencia es que las consultas a los routers mediante este protocolo no inyectan una carga de procesado a la CPU, ya que solo se limita a escribir o leer las ya presentadas MIBs.

SNMP no es capaz de proporcionar datos en profundidad como pueden ser origen/destino del tráfico, aplicaciones que se utilizan, ToS o calidad de servicio.

CAPÍTULO 2. PLANIFICACIÓN DEL ESCENARIO DE PRUEBAS

El objetivo marcado para este proyecto es el estudio de la herramienta `pmacct`, documentarlo y entender todo el potencial que ofrece y a su vez analizar las deficiencias del mismo, para una posible incorporación docente en el estudio de la monitorización de redes. A continuación presentamos la herramienta `pmacct` y todo el software desplegado para la construcción de nuestro sistema.

2.1. pmacct

`pmacct` (*Promiscuous mode IP Accounting package*) [7] es un proyecto que empezó a desarrollarse a inicios del año 2003 y que sigue en continua evolución, el cual, su principal desarrollador ha sido Paolo Lucente.

Para la realización del proyecto, se ha escogido la versión más reciente "pmacct-0.14.2", aconsejada por el propio Paolo [14] ya que es a la que mejor soporte podría proporcionar.

`pmacct` es un pequeño conjunto de herramientas pasivas que nos vas a permitir monitorizar redes, con el fin de medir, clasificar, cuantificar, agregar y exportar tráfico IPv4 e IPv6. Sus principales características son:

- Es adecuado para:
    - ISP (Internet Service Provider) → Empresa que proporciona conexión a internet a sus usuarios. En España por ejemplo tenemos Movistar, ONO, Jazztel entre otras. Un ejemplo de uso de este software podría ser para conocer qué recursos son más utilizados por los clientes y con ello poner en marcha prácticas de mejora en el sistema.
    - IXP (Internet Exchange Point) → Infraestructura física a través de la cual los ISP intercambian tráfico entre sus redes, sufragada proporcionalmente entre sus usuarios. Por ejemplo, el punto neutro de Barcelona (Catnix) y el de Madrid (ESPANIX).
    - CPD (Centro de procesado de datos) → Ubicación donde se concentran los recursos necesarios para el posterior procesado de la información. Un ejemplo de uso en este sistema en una empresa privada podría ser conocer qué operaciones están realizando sus trabajadores y como están usando los recursos.
- Funciona sobre los SO Linux, BSD, Solaris.

- Da soporte para IPv4 e IPv6.
- Permite generar informes de red a través de:
    - libpcap (library packet capture) [15] → Implementación de la aplicación “pcap” para sistemas Unix de captura de paquetes que se transmiten por la red. Ejemplos de softwares que lo utilizan: tcpdump, Wireshark.
    - NetFlow v1/v5/v7/v8/v9
    - sFlow
- Nos permite gestionar el contenido de los informes de red mediante una serie de backends internos o con las tablas de memoria, MySQL, PostgreSQL, SQLite, BerkeleyDB y archivos de texto plano.
- Permite exportar datos a los colectores remotos mediante NetFlow v5/v9 y sFlow v5 e IPFIX.
- Incorpora un daemon BGP para tareas de encaminamiento entre sistemas autónomos.
- Incorpora un daemon IS-IS para tareas de encaminamiento de rutas internas.
- Puede insertar datos en herramientas externas como por ejemplo, Cacti, Net-SNMP, MRTG

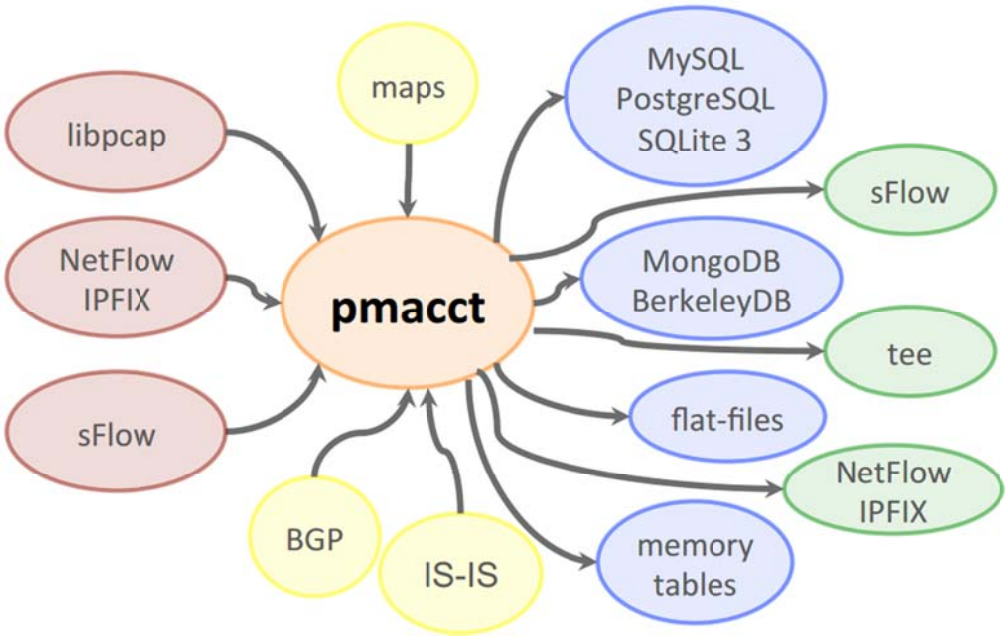


Fig. 2.1 Componentes de `pmacct`

2.1.1. Descripción del software desplegado

En esta sección vamos a introducir cual es el diseño de red sobre el que se ha trabajado y que tecnologías se han utilizado para ello. En posteriores capítulos se explicará con más detalle dichas tecnologías y las configuraciones que se han llevado a cabo para el correcto funcionamiento del sistema.

2.1.1.1 SO utilizado

Una de las limitaciones del software `pmacct` es que no se ejecuta sobre SO Windows. Por ello se ha decidido utilizar el SO Ubuntu (basado en Debian) ya que es el sistema que se despliega en los laboratorios de la escuela EETAC para asignaturas como “Laboratorio de Redes”. La versión utilizada ha sido Ubuntu 11.10:

```
root@alfonso-K53SD:/home/alfonso# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 11.10
Release:        11.10
Codename:       oneiric
root@alfonso-K53SD:/home/alfonso#
```

Fig. 2.2 Consulta en la Shell para obtener la versión del SO

2.1.1.2 Escenario a desarrollar

Es un sistema formado por seis routers (en tres sistemas autónomos) y una máquina terminal, los cuales van a poder intercambiar información mediante encaminamiento interno y externo. En ellos se va a recolectar información sobre el tráfico que se transmite por la red gracias a agentes NetFlow y SNMP que será enviada a la máquina gestora y ésta para el procesado de la información la guardará en un sistema de almacenamiento.

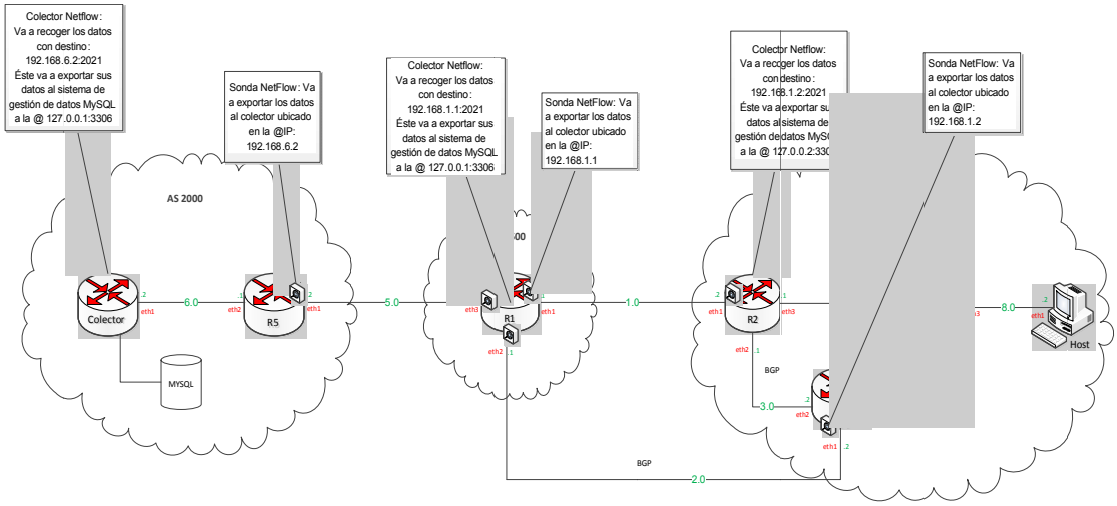


Fig. 2.3 Esquema de red a desarrollar

2.1.1.3 Software de virtualización

En el ámbito de la informática, se define virtualización como la creación a través de software o aplicación de una versión virtual de algún tipo de recurso tecnológico como un sistema operativo, dispositivo de almacenamiento entre otros.

Se optó por desarrollar el escenario haciendo uso de máquinas virtuales utilizando la aplicación Oracle VM Virtualbox [16]. Es un software de virtualización que podemos obtener del Centro de Software de Ubuntu. Esta aplicación nos va a permitir instalar sistemas operativos adicionales, conocidos como "sistemas invitados" sobre el sistema operativo nativo de nuestra máquina conocida como "anfitriona", obteniendo de cada uno de ellos su propio entorno virtual.

Se van a crear un total de siete VM (Virtual Machines). Seis de ellas van a realizar la función de router y una de ellas de máquina terminal. En próximos capítulos, se presentará con detalle cómo configurar una VM para que conecte con las demás.

La versión que se ha utilizado de `VirtualBox` es la siguiente:

```
root@alfonso-K53SD:/home/alfonso# dpkg-query -W virtualbox
virtualbox      4.1.2-dfsg-1ubuntu1.1
root@alfonso-K53SD:/home/alfonso#
```

Fig. 2. 4 Consulta desde la Shell para obtener versión de software `virtualBox`

A continuación se presenta una captura del Administrador de `VirtualBox`, con todas las VM ya creadas (Figura 2.5).

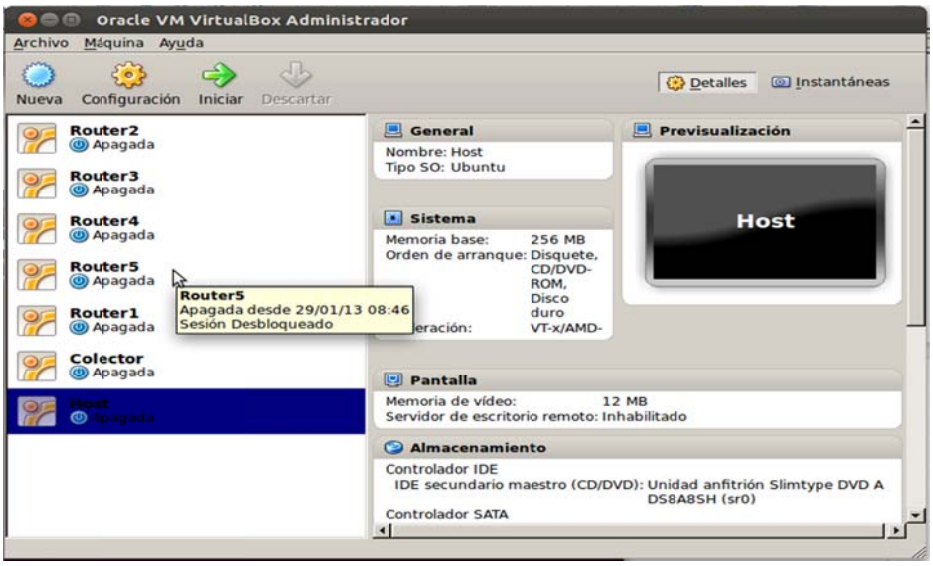


Fig. 2.5 Colección de VM creadas en el sistema

2.1.1.4 Software de encaminamiento

Se define encaminamiento como la función de buscar un camino óptimo o mejor ruta, normalmente asociado a un router, para permitir la interoperabilidad entre pares de nodos de red.

Para nuestro sistema, se ha optado por utilizar el software quagga [17], ya que es la usada en la EETAC para introducir conceptos de encaminamiento.

quagga es un software avanzado que proporciona los protocolos de encaminamiento basados en TCP/IP, como RIP, OSPF o BGP. Para nuestro estudio, se van a utilizar tanto protocolos de encaminamiento interno (IGP, *Internal Gateway Protocol*) como protocolos de encaminamiento externo (EGP, *External Gateway Protocol*).

- IGP → Protocolos de encaminamiento utilizados dentro del mismo AS (*Autonomous System*). Un AS es un conjunto de redes IP que tienen una política de rutas propia e independiente y realiza su propia gestión del tráfico que fluye entre él y los demás AS vecinos. En nuestro estudio se ha utilizado:
    - OSPF (*Open Shortest Path First* - RFC2328) → Protocolo interior que soporta grandes redes y escalables que se creó como solución a RIP (*Routing Information Protocol* - RFC1058), ya que entre otras desventajas, tenía una limitación de trabajar con redes con un máximo de quince saltos.
- EGP → Protocolos de encaminamiento utilizado para intercambiar información de encaminamiento entre AS. Se ha utilizado:
    - BGP (Border Gateway Protocol - RFC1771) → Protocolo de encaminamiento entre AS. Su principal función es la de intercambiar información de encaminamiento con otros sistemas BGP, con el fin de aprender la ruta hacia otras redes. Esta información, entre otras, incluirá una lista de los ASs que se tienen que atravesar para llegar al destino final, conocido como AS_PATH.

En próximos capítulos se explicarán con detalle, los archivos de configuración que se han creado para interconectar todas las VM. La versión utilizada ha sido (Figura 2.6):

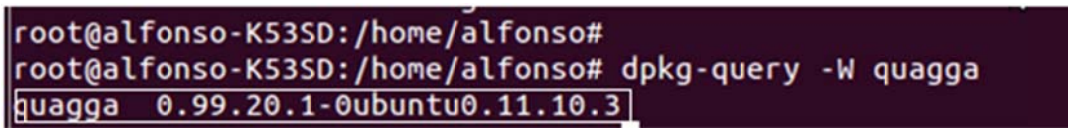


Fig. 2.6 Consulta desde la Shell para obtener versión de software quagga

Esta versión es la que se obtiene, descargando la aplicación desde el Centro de Software de Ubuntu.

2.1.1.5 Software de recolección de información

De las dos alternativas presentadas en el capítulo anterior, SNMP o NetFlow, se ha utilizado NetFlow, ya que es una de las aplicaciones que complementa a `pmacct`.

`pmacct` incluye un daemon NetFlow, llamado nfacctd que actuará como colector y un plugin adicional llamado nfprobe que actuará como agente/sonda NetFlow.

En los routers donde el software `pmacct` no se ejecuta, se ha utilizado como agente NetFlow, la aplicación fprobe (Netflow Probe) [18]. Es una herramienta basada en libpcap que se encarga de recoger datos del tráfico de red en la/s interfaces definidas y emitirlo como flujos NetFlow al colector especificado.

La versión utilizada de este agente NetFlow es (Figura 2.7):

Fig. 2.7 Consulta desde la Shell para obtener versión de software fprobe

2.1.1.6 Software de almacenamiento de información

En este punto se va a describir qué SGBD (Sistema de gestión de Base de Datos) se ha utilizado, para completar las tareas de monitorización.

De las mencionadas en el punto 2.1, entre las que se encuentran MySQL, PostgreSQL, BerkeleyDB o tabla de memoria interna, se ha elegido MySQL, entre otros motivos, por dar soporte visual a la aplicación con el complemento "MySQL Query Browser".

`pmacct` presenta un plugin llamado mysql que va a permitir conectar el software `pmacct` con el sistema de gestión de base de datos MySQL [19]. Creando una base de datos llamada `pmacct` se va a poder guardar la información gestionada.

La versión utilizada en nuestro estudio es la presentada en el Centro de Software de Ubuntu (Figura 2.8):

```
mysql> select version();
+-------------------------+
| version()               |
+-------------------------+
| 5.1.66-0ubuntu0.11.10.3 |
+-------------------------+
1 row in set (0.00 sec)
```

Fig. 2.8 Consulta desde la Shell para obtener versión de software MySQL

A continuación se presenta una captura donde se ejemplifica el uso del complemento “MySQL Query Browser” y su comparativa desde la Shell (Figura 2.9):

Fig. 2.9 Consulta de tablas MySQL desde la Shell y MySQL Query Browser

CAPÍTULO 3. CONFIGURACIÓN DEL SISTEMA

En este capítulo se van a presentar los archivos de configuración que se han llevado a cabo para el correcto funcionamiento de nuestro escenario.

3.1 Configuración de máquinas virtuales

Cómo se documentó en el anterior capitulo, el software de virtualización utilizado ha sido Oracle VM Virtualbox. En este apartado se va a describir la configuración que se ha llevado a cabo para permitir la conectividad a nivel de encaminamiento con las demás máquinas virtuales. Se va a utilizar como ejemplo la máquina virtual que se ha definido con el nombre Router 1. Se podrá encontrar el manual de cómo crear una máquina virtual y la configuración del resto de máquinas virtuales en el anexo I.

Se abre un nuevo terminal con permisos de superusuario y accedemos a la aplicación (Figura 3.1):

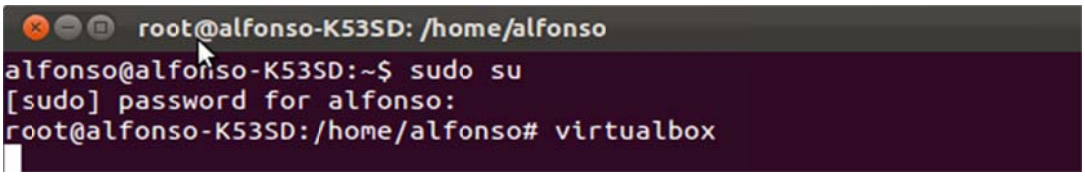


Fig. 3.1 Acceso desde la Shell a la aplicación `VirtualBox`

Se abrirá la siguiente ventana (Figura 3.2). En ella se presentan todas las capacidades administrativas de `VirtualBox`, como puede ser, crear una nueva máquina virtual `Nueva`, configuración de la máquina virtual `Configuración` o arrancar la máquina virtual `Iniciar`.

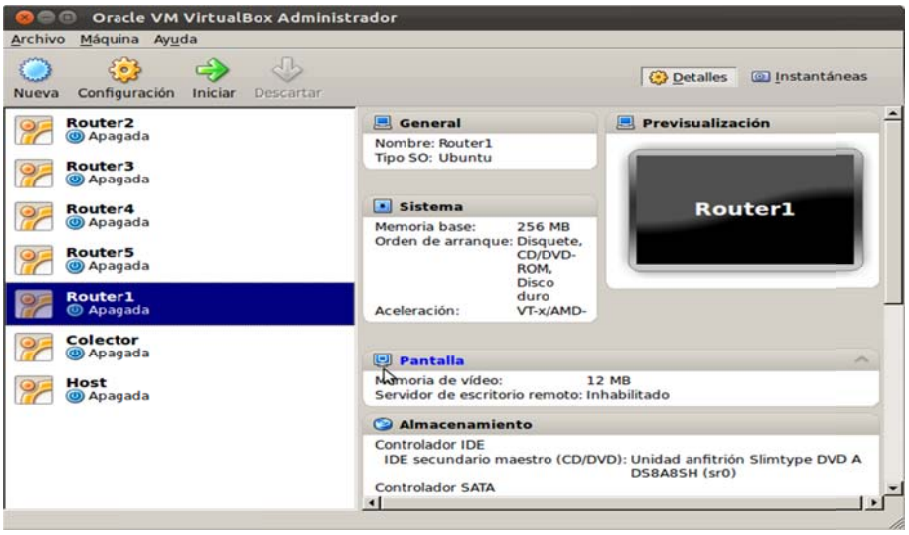


Fig. 3.2 Panel de administración `VirtualBox`

Se selecciona la máquina virtual, en la cual se quieren realizar tareas administrativas y pulsamos el botón `Configuración` (Figura 3.3).

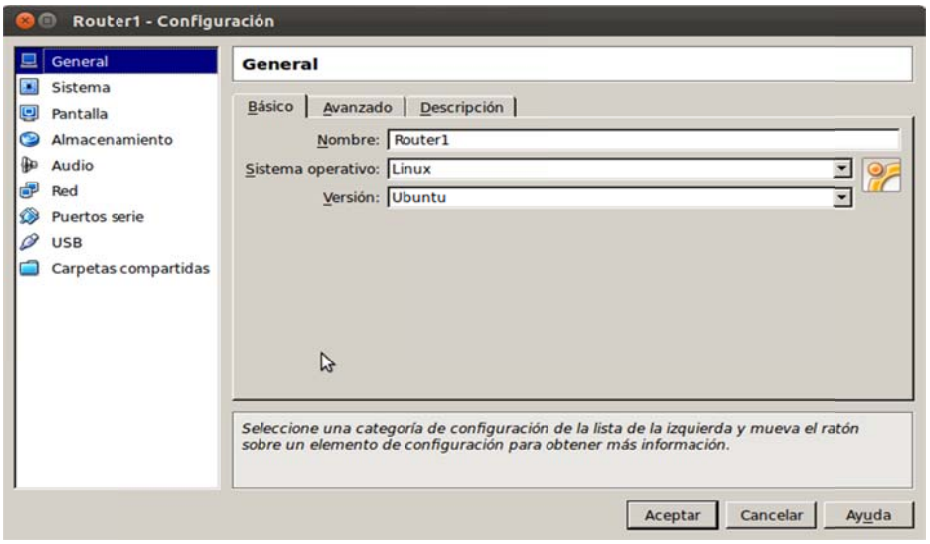


Fig. 3.3 Panel de configuración máquina virtual Router 1

Se presentan una serie de secciones como `General`. En ella se presenta información básica del sistema como el nombre de la máquina virtual, sistema operativo sobre el que está funcionando o la versión del mismo. Destacamos que la propia aplicación presenta una breve descripción de que es cada campo pudiéndola obtener pasando el ratón sobre el elemento.

Se va realizar un especial énfasis sobre la sección `Red` (Figura 3.4), ya que la correcta configuración de la misma es la que nos va a permitir la interoperabilidad con las demás máquinas virtuales o en su defecto con la máquina anfitriona.

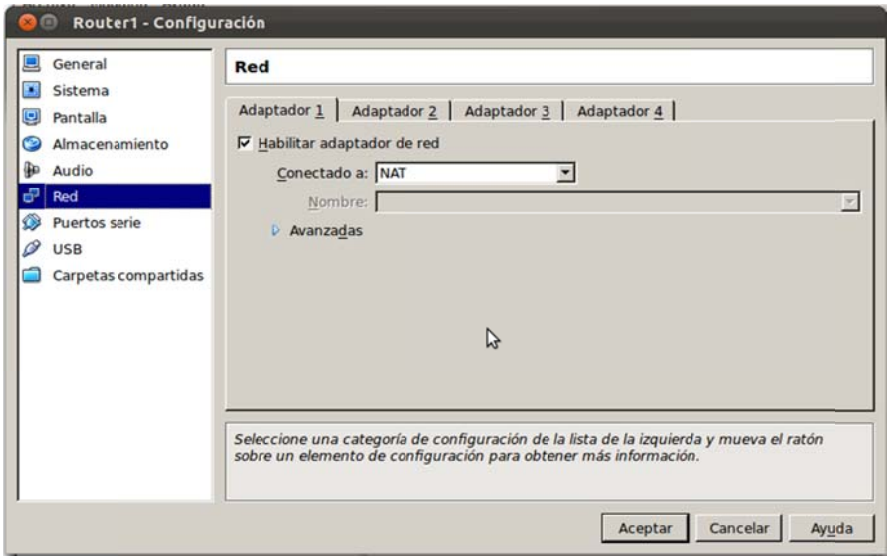


Fig. 3.4 Panel de configuración sección Red

VirtualBox está limitado a proporcionar cuatro interfaces virtuales, presentadas como Adaptador 1, Adaptador 2… que se corresponderán a eth0, eth1 … respectivamente. Por defecto, solo se encuentra habilitado el Adaptador 1. Para habilitar el resto de interfaces, basta con pulsar el adaptador que se desea activar y hacer click en la pestaña Habilitar adaptador de red.

VirtualBox ofrece diferentes posibilidades de conexión de red:

- No conectado → Muestra un adaptador de red pero sin conexión (estado desconectado).
- NAT (Network Address Translation) → Permite al huésped navegar por internet sin necesidad de configurar el SO.
- Adaptador Puente → Simula una conexión física real a la red, asignando una IP al SO huésped. Va a permitir la conexión con la máquina anfitriona.
- Red Interna → Similar a Adaptador Puente con la salvedad de que la conexión solo se va a poder realizar con máquinas virtuales conectadas en la misma red interna.

Para nuestro sistema se debe escoger Red Interna. Se presenta la configuración final de Router 1- Router 2- Router 3 (Figura 3.5) directamente conectados como se ilustra en la figura 2.3.

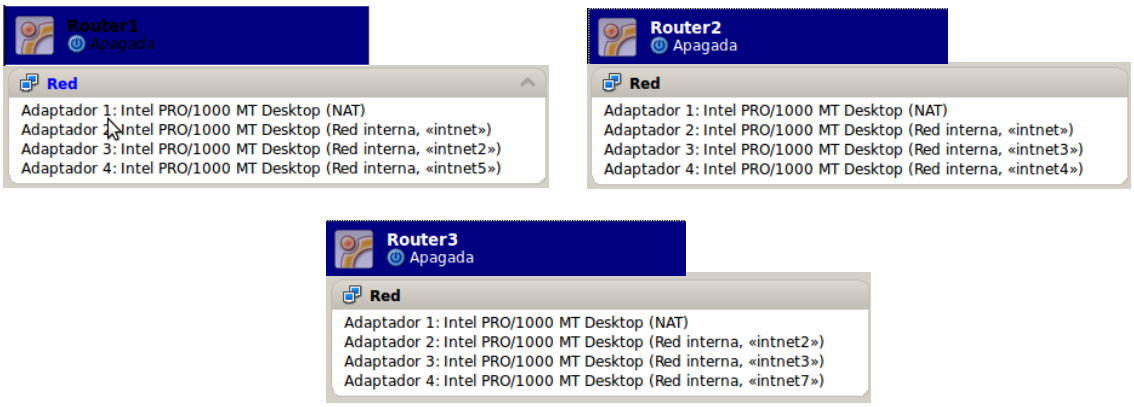


Fig. 3.5 Configuración final VirtualBox sección red.

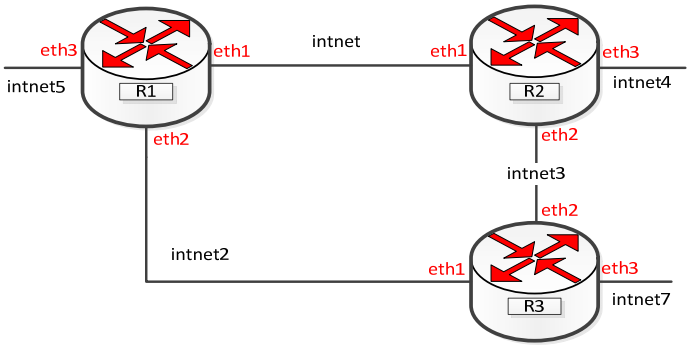


Fig. 3.6 Representación de la configuración VirtualBox en esquema de red

En este apartado se ha definido como se van a interconectar los routers. Por sí solos no van a conocer la existencia de los demás ni van a poder intercambiar información. Para ello se procede a configurar cada una de las interfaces con el software de encaminamiento.

3.2 Configuración del software de encaminamiento

En el apartado 2.1.1.4 se presenta el software de encaminamiento a utilizar. En este apartado se describe la configuración que se ha llevado a cabo y la demostración de su correcto funcionamiento. La descripción de instalación del software y configuración de todas las máquinas, se puede encontrar en el anexo II.

Una máquina terminal o host no es capaz de realizar funciones de encaminamiento, tareas dedicadas a un router o switch. En nuestro caso, el primer paso, es activar dicha función (Figura 3.7).

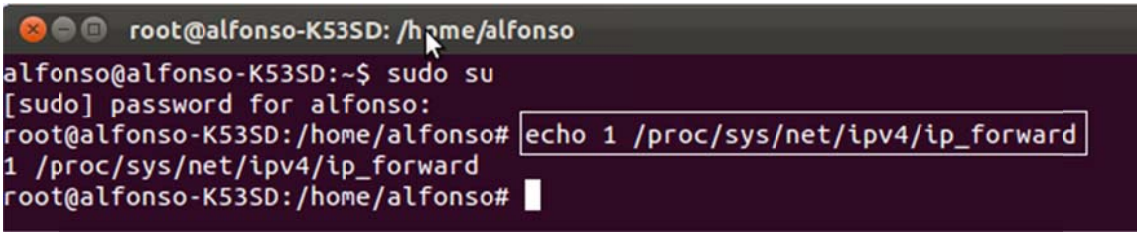


Fig. 3.7 Activación de la función de enrutamiento desde la Shell.

quagga presenta una serie de daemons, cada uno de ellos con su respectivo archivo de configuración (Figura 3.8).

Fig. 3.8 Consulta de daemons activos / archivo de configuración

En este ejemplo, extraído de R2, se activan los daemons, zebra, bgpd y ospfd. El daemon zebra, nos va a permitir asignar IP estáticas. Con esto se consigue tener conectividad con los routers directamente conectados pero no tener conocimiento del resto de routers. Para ello, hay que hacer uso de protocolos de encaminamiento dinámico que son BGP y OSPF. Se presentan los archivos de configuración creados en R2 (Figura 3.9) y la tabla de encaminamiento aprendida por el mismo (Figura 3.10). Una tabla de encaminamiento presenta las rutas aprendidas de diferentes nodos en una red.

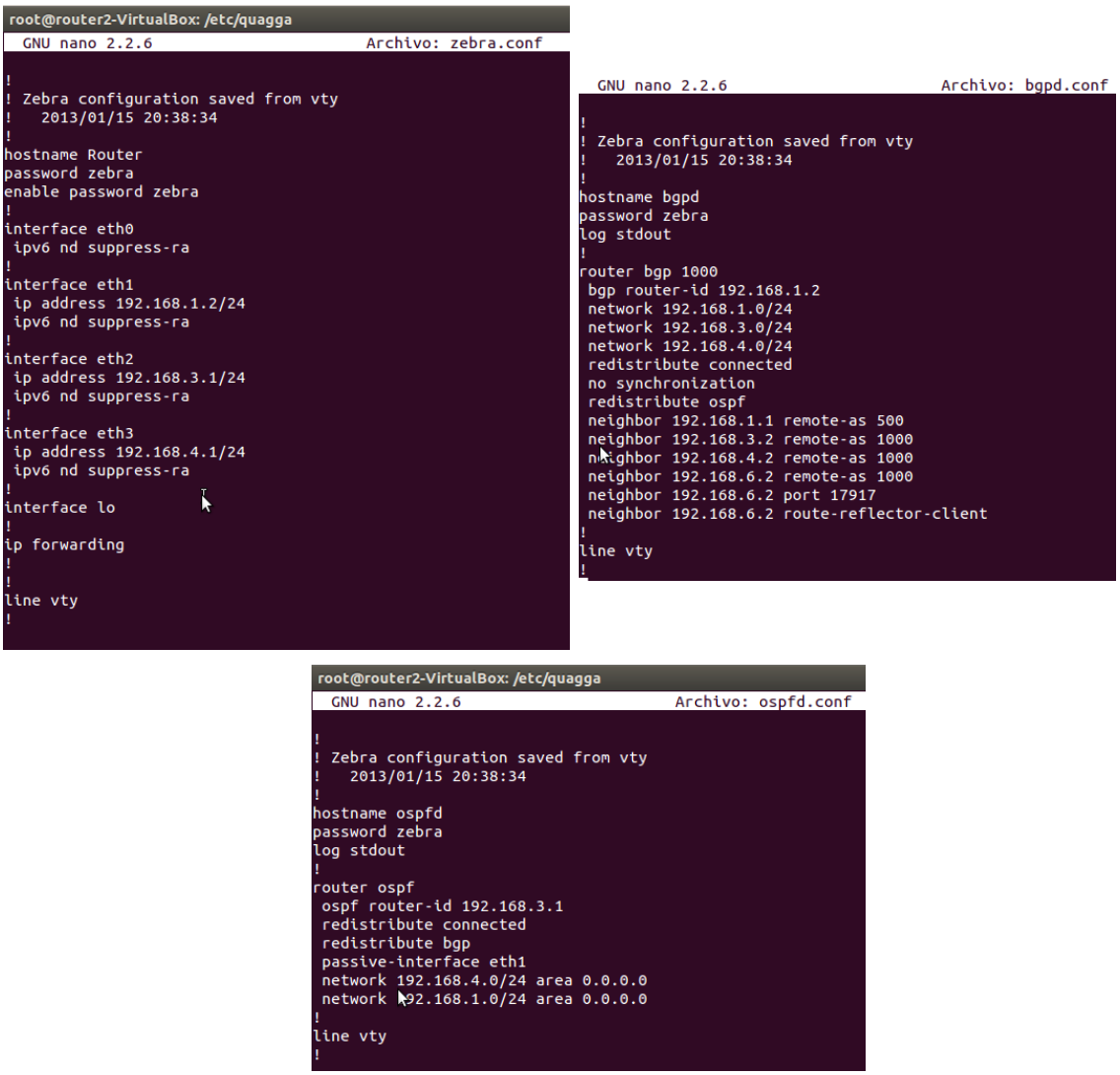


Fig. 3.9 Archivos de configuración R2

En el archivo de configuración `zebra.conf` se presenta la asignación de IP estática para cada una las interfaces de red. La interfaz eth0 no está configurada, ya que se reserva para tener acceso a navegación web y es configurada con `VirtualBox` como NAT. Éste le asignará una dirección IP automáticamente via DHCP.

R2 forma parte de un AS donde operan más de un router; por ello hay que configurar un protocolo de encaminamiento interior: OSPF. Se asigna un identificador de router, que será cualquiera de las IPs asignadas a una interfaz `ospf router-id`. Los comandos `redistribute` servirán para redistribuir rutas a otro protocolo de encaminamiento. En este caso, se redistribuyen todas las rutas directamente conectadas `redistribute connected` y las rutas aprendidas vía BGP `redistribute bgp`. La configuración `passive-interface eth1` nos proporciona que el router por dicha interfaz va a actuar de forma pasiva respecto a este protocolo, es decir, va a escuchar tráfico por ésta interfaz pero no va a establecer conexión OSPF. El comando `network` servirá para definir que redes se van a publicar mediante OSPF y `area` es un

parámetro que introduce OSPF para reducir las tablas de encaminamiento. En un área se aplica el protocolo OSPF como si se tratase de una red independiente, es decir, los routers que forman un área solo van a aprender las redes anunciadas dentro del área y no la totalidad de la red.

R2 es un router frontera. Intercambia información con routers pertenecientes a otro AS. Para que la comunicación sea posible se ha de configurar un protocolo de encaminamiento externo: BGP. Se define a qué AS pertenece router `bgp 1000` y un identificador de router. Se anuncian las redes a las que pertenece `network x.x.x.x` y la relación de vecindad con los routers con los que se quiere abrir una sesión BGP `neighbor 192.168.1.1 remote-as 500`. En este caso, se define que se quiere establecer una sesión BGP con el R1 directamente conectado a través de la IP 192.168.1.1 y que pertenece al AS 500. El último caso, se explicará con más detalle, en la configuración del daemon bgp introducido por `pmacct`.

```
router2-VirtualBox# sh ip bgp route
% Command incomplete.
router2-VirtualBox#
router2-VirtualBox# sh ip bgp
BGP table version is 0, local router ID is 192.168.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i10.0.2.0/24     192.168.3.2              1    100      0 ?
*                  192.168.1.1              1             0 500 ?
*>                 0.0.0.0                  1         32768 ?
* i192.168.1.0     192.168.7.2             20    100      0 ?
*                  192.168.1.1              0             0 500 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
*> 192.168.2.0     192.168.4.2             20         32768 ?
* i                 192.168.3.2              0    100      0 i
*                  192.168.1.1              0             0 500 i
* i192.168.3.0     192.168.3.2              0    100      0 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
* i192.168.4.0     192.168.7.2             20    100      0 ?
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
* i192.168.5.0     192.168.2.1              0    100      0 500 i
*>                 192.168.1.1              0             0 500 i
* i192.168.6.0     192.168.2.1                   100      0 500 2000 i
*>                 192.168.1.1                            0 500 2000 i
*> 192.168.7.0     192.168.4.2             20         32768 ?
* i                 192.168.3.2              0    100      0 i
* i192.168.8.0     192.168.7.2             20    100      0 ?
*>                 192.168.4.2             20         32768 ?

Total number of prefixes 9
```

Fig. 3.10 Tabla encaminamiento BGP aprendida por R2

En la figura se presentan las redes aprendidas por R2 `Network` y los diversos caminos que tiene para llegar a ellas `Next Hop` y el número de sistemas autónomos que tiene que travesar para llegar a ellas `Path`.

A continuación, se demuestra el correcto funcionamiento del sistema, estableciendo una conexión extremo a extremo de la red diseñada (Figura 3.11).

```
root@host-VirtualBox:/home/host# traceroute 192.168.6.2
traceroute to 192.168.6.2 (192.168.6.2), 30 hops max, 60 byte packets
 1  router4-VirtualBox.local (192.168.[illegible].1)  0.323 ms  0.480 ms  0.322 ms
 2  192.168.7.1 (192.168.7.1)  0.690 ms  3.687 ms  3.600 ms
 3  192.168.2.1 (192.168.2.1)  3.495 ms  3.447 ms  3.344 ms
 4  192.168.5.2 (192.168.5.2)  7.312 ms  7.260 ms  7.166 ms
 5  192.168.6.2 (192.168.6.2)  7.076 ms  6.952 ms  7.437 ms
root@host-VirtualBox:/home/host#
```

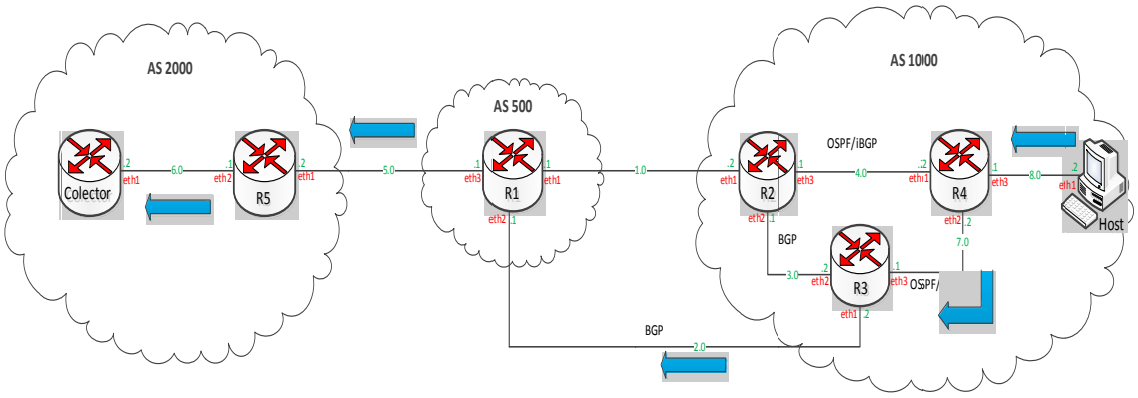


Fig. 3.11 Traceroute de la máquina Host a Colector

Una vez comprobado el correcto funcionamiento del encaminamiento, se procede a configurar aplicación `pmacct`.

3.3 Configuración del software `pmacct`

`pmacct` basa su funcionamiento en una serie de directivas que se pueden encontrar en "http://wiki.pmacct.net/OfficialConfigKeys". En el próximo apartado se explicará con detalle cual es su funcionamiento basándonos en el archivo de configuración creado en máquina "Colector" y "R5".

`pmacct` presenta una serie de daemons para generar informes de red:

- `pmacctd` (Promiscuous mode accounting daemon) → **Basado en libpcap**. Es una de las aplicaciones más extendidas en la materia de recolección de paquetes, ejemplo de ello, pueden ser `tcpdump, wireshark o snort.`

- `nfacctd` (NetFlow accounting daemon) → Basado en NetFlow v1/v5/v9, va a permitir la creación de un colector NetFlow.

- `sfacctd` (Sflow accounting daemon) → Basado en Sflow v5 va a permitir la creación de un colector sFlow.

`pmacct` presenta una serie de daemons para configurar la estructura de red:

- BGP → Para activar esta función se requiere que en el proceso de configuración y compilación del software se habilite el uso de múltiples hilos de ejecución. Puede encontrar toda la información en el anexo XII. La activación va a permitir que el dispositivo escuche conexiones BGP de forma pasiva, es decir, no va a establecer una conexión con un interlocutor remoto, pero espera a conexiones entrantes.

`pmacct` presenta una serie de plugins tanto para el almacenamiento de datos como la generación de los mismos:

- Memory → permite el uso de una tabla de memoria como backend que va a poder ser consultada a través de la aplicación cliente `pmacct`.

- Print → imprime los datos recogidos en ficheros de texto plano.

- Mysql, pgsql, sqlite3 → solo van a poder ser utilizados si se han configurado y compilado en el proceso de instalación. Van a permitir el uso de sus respectivas aplicaciones.

- Nfprobe → va a permitir la actuación como agente Netflow.

- Sfprobe → va a permitir la actuación como agente Sflow.

`pmacct` presenta una serie de scripts para la obtención de valores definidos por el usuario en archivos.txt como por ejemplo, lista de puertos, lista de redes entre otros.

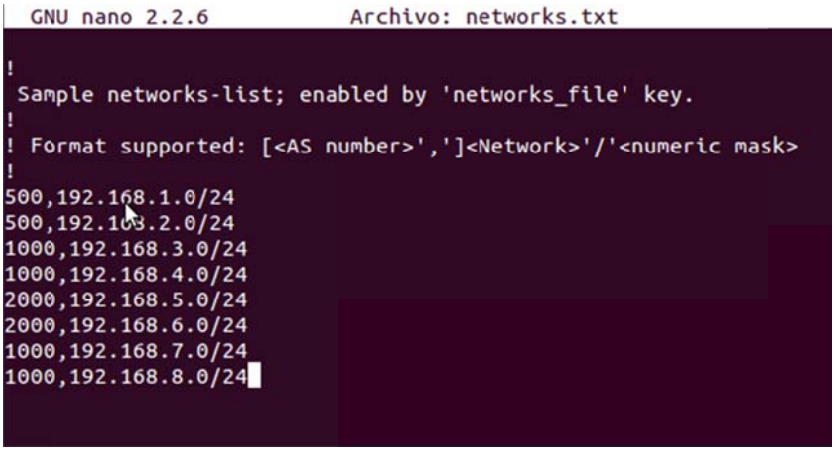


Fig. 3.12 Fichero networks.txt

El principio de funcionamiento es el siguiente. Consulta el fichero y se va a asociar la dirección IP recibida con un número de AS al que pertenece.

Nuestro sistema, va a trabajar sobre los daemons `nfacctd, bgp` y los plugins `mysql y nfprobe`. Se va a implementar en cada AS, siendo “Colector”, “R1”

y “R2” los dispositivos a configurar. El esquema final definido por AS es el que se presenta en la Fig. 3.13:

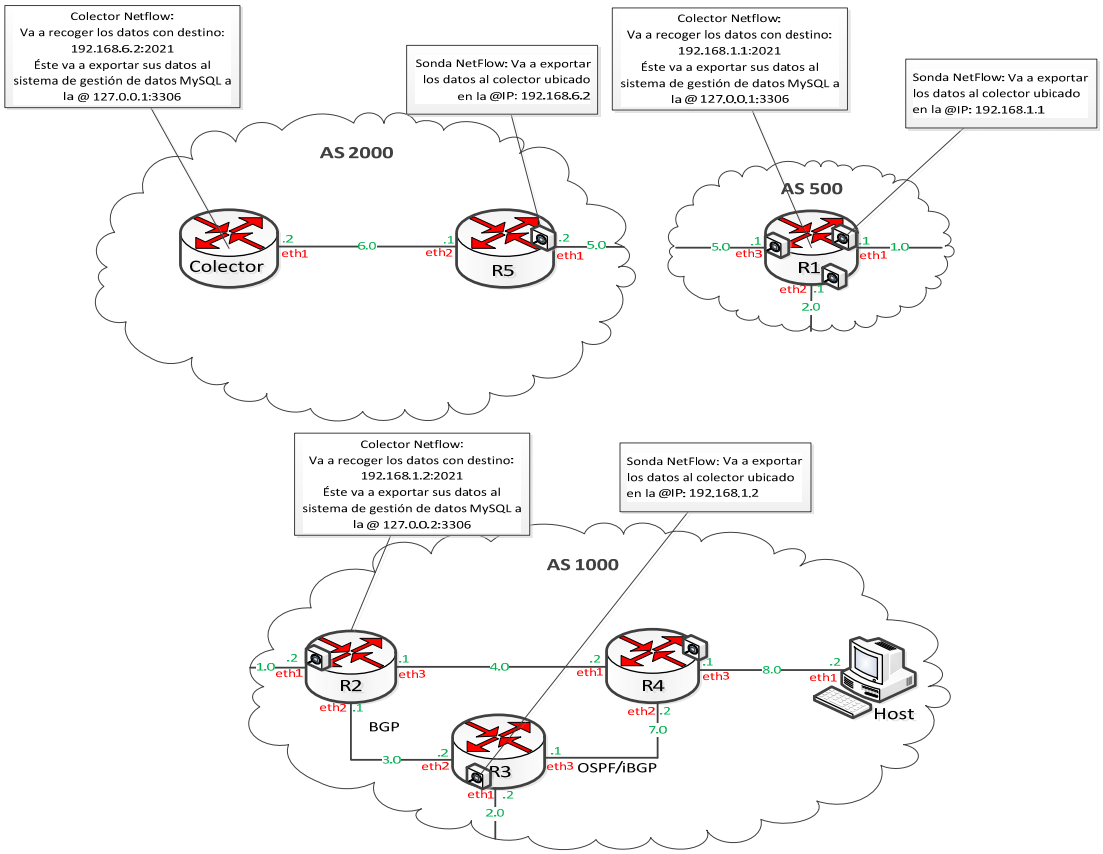


Fig. 3.13 Esquema final de la red

Para probar todas las soluciones posibles se ha definido la siguiente estructura (Figura 3.14):

AS	Dispositivo	Encaminamiento	Gestión contenido
2000	Colector	Daemon BGP Pmacct →Escucha peticiones puerto 17929	Daemon Nfacctd Pmacct → Colector NetFlow 192.168.6.2:2021
	R5	BGP – OSPF software Quagga	Plugin nfprobe Pmacct →Escucha en eth1 y envía a 192.168.6.2:2021
500	R1	BGP software Quagga. Fichero networks.txt	Daemon Nfacctd Pmacct → Colector NetFlow 192.168.1.1:2021 Agente NetFlow fprobe → Escucha en eth1,eth2,eth3 y envía a colector 192.168.1.1:2021
1000	R2	BGP – OSPF software Quagga. Fichero networks.txt	Daemon Nfacctd Pmacct → Colector NetFlow 192.168.1.2:2021 Plugin nfprobe Pmacct → Escucha en eth1 y envía a colector 192.168.1.2:2021
	R3	BGP – OSPF software Quagga	Agente NetFlow fprobe → Escucha en eth1 y envía a colector 192.168.6.2:2021
	R4	OSPF software Quagga	Agente NetFlow fprobe → Escucha en eth3 y envía a colector 192.168.6.2:2021

Fig. 3.14 Definición de la estructura `pmacct`

El colector NetFlow ubicado en la máquina Colector va a recibir los registros NetFlow generados por la máquina R5. El plugin `nfprobe` presenta una deficiencia en cuanto a la obtención de valores referentes a enrutamiento (Figura 1.6). Los valores como `AS origen, AS destino` entre otros van a ser marcados con valor 0. `pmacct` presenta una solución que es recoger estos valores de la información proporcionada por el daemon BGP configurado en esta máquina.

El colector NetFlow ubicado en la máquina R1 recibirá los registros NetFlow generados en eth1, eth2 y eth3 por el software fprobe. Al igual que el plugin nfprobe, los valores referentes a enrutamiento serán marcados como 0. `pmacct` presenta otra solución al respecto y es obtener esta información del fichero `networks.txt` (Figura 3.12).

3.3.1. Configuración de la máquina Colector – R5

En este apartado, se va a presentar tanto el archivo de configuración creado en máquina Colector (Figura 3.15) como en R5 (Figura 3.16) y cuál es el funcionamiento de cada uno de los parámetros presentados Los archivos de configuración del resto de máquina se presentarán en el anexo XII.

```
  GNU nano 2.2.6                Archivo: nfacctd1.conf                Modificado

!
! nfacctd configuration example
!
! Did you know CONFIG-KEYS contains the detailed list of all configuration keys
! supported by 'nfacctd' and 'pmacctd' ?
!
! debug: true
daemonize: false

!PRIMITIVAS --> SE VAN A RECOGER LOS SIGUIENTES DATOS!
aggregate: src_host, dst_host, src_as, dst_as, src_port, dst_port, as_path, peer_src_ip, peer_dst_ip,
peer_src_as, peer_dst_as, flows

!PLUGINS A ACTIVAR --> se activa plugin mysql, ya que es la DB donde vamos a guardar los registros recolectados
plugins: mysql

!CONFIGURACIÓN DB MYSQL!
sql_db: pmacct
sql_table: acct_bgp2
sql_history: 5m
sql_refresh_time: 300
sql_table_version: 4
sql_table_type: bgp
sql_host: 127.0.0.1
sql_user: root
sql_passwd: bluesky88
sql_history_roundoff: m
sql_optimize_clauses: true

!CONFIGURACIÓN COLECTOR NETFLOW!

nfacctd_port: 2021
nfacctd_ip: 192.168.6.2
nfacctd_as_new: bgp
nfacctd_net: bgp
nfacctd_time_new: true

!CONFIGURACIÓN DAEMON BGP!

bgp_daemon: true
bgp_daemon_ip: 192.168.6.2
bgp_daemon_port: 17917
bgp_daemon_max_peers: 6
bgp_daemon_msglog: true
bgp_peer_src_as_type: bgp
bgp_src_as_path_type: bgp
```

Fig. 3.15 Archivo de configuración del daemon Nfacctd

En el archivo de configuración `nfacctd1.conf`, se presentan una serie de primitivas `aggregate` que van a ser los valores recogidos de los registros Netflow, como, IP origen `src_host`, IP destino `dst_host`, AS origen `src_as`, AS destino `dst_as` entre otras. El resto de primitivas proporcionadas por `pmacct` se pueden encontrar en "http://wiki.pmacct.net/OfficialConfigKeys".

Se presentan los valores de configuración para exportar los datos recibidos en el colector al sistema de gestión de base de datos que en nuestro caso es MySQL. Se hace una llamada a la base de datos `pmacct` y se especifica en que tabla se deben transcribir los datos `acct_bgp2`. Las entradas a la tabla se van a dividir en entradas de cinco minutos `sql_history:5m` valores representados como `stamp_inserted (tiempo de entrada)` y `stamp_updated (tiempo en que se han transcrito los datos)`. Cada cinco minutos se van a realizar consultas a la tabla, es decir, se van a enviar los datos almacenados en el colector NetFlow `sql_refresh_time:300`. Se define la tabla como tipo bgp, ya que es necesario almacenar información referente a campos de encaminamiento como AS origen, AS destino, entre otros.

Se definen los parámetros para la configuración del colector NetFlow, donde se indica que va a escuchar registros enviados al puerto `2021` y dirección IP `192.168.6.2`. Son campos que habrá que tener en cuenta a la hora de configurar el agente NetFlow. Tomaremos como ejemplo el archivo de configuración de R5 (Figura 3.16), dispositivo donde se configura un agente NetFlow. Los campos `nfacctd_as_new - nfacctd_net` van a obtener la información proporcionada por el daemon bgp ya que nuestro agente NetFlow se encuentra limitado en estos aspectos. Por último, el campo `nfacctd_time_new` va a servir para eliminar las marcas de tiempo incluidas en la cabecera de NetFlow y construir otras nuevas que será, tiempo de entrada en el colector. Esto nos va a servir para pruebas futuras, como va a ser la construcción de matrices de tráfico.

Se describen los parámetros para la configuración del daemon bgp, donde se define, la dirección IP y puerto donde va a escuchar `192.168.6.2:17917`. Estos valores hay que tenerlos en cuenta a la hora de configurar los dispositivos donde se está ejecutando el daemon bgp a cargo del software `quagga` (Figura 2.9).

Una de las deficiencias que presenta el daemon bgp es que solo entiende de iBGP (internal BGP), protocolo BGP dentro del mismo AS. Al configurar las máquinas donde se ejecuta aplicación `quagga`, hay que indicar que máquina colector forma parte del mismo AS, como se muestra en los archivos de configuración de R2 (Figura 2.9).

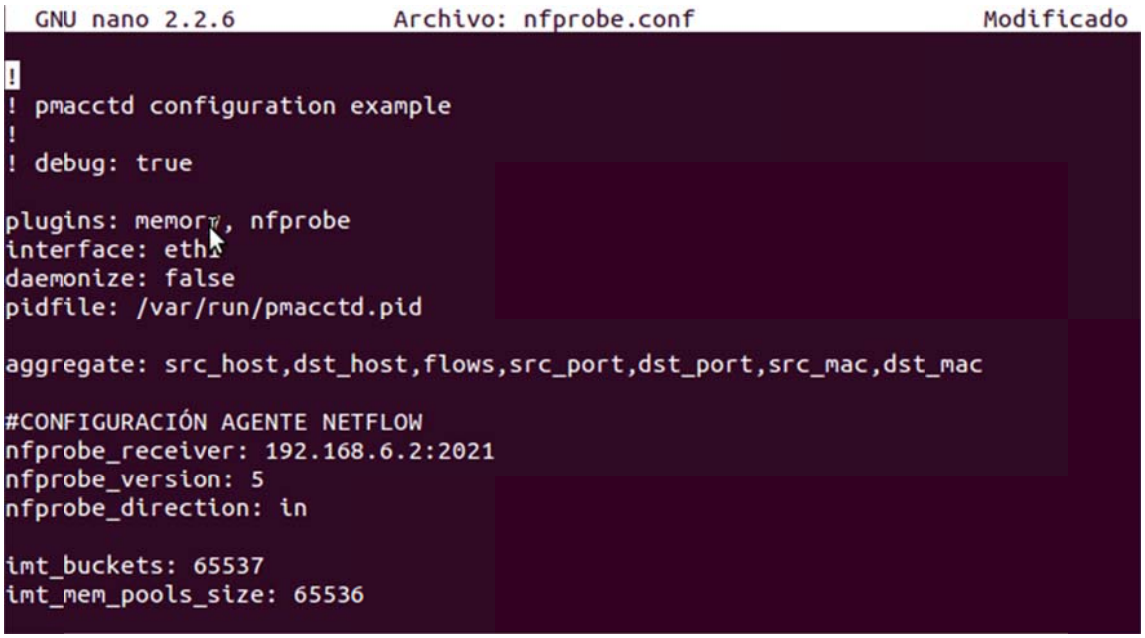


Fig. 3.16 Configuración del agente NetFlow en R5

Se define que el agente NetFlow escuche el tráfico de entrada en la interfaz eth1 `nfprobe_direction – interface`. Se indica que el colector NetFlow escuchará registros en la dirección IP:puerto 192.168.6.2:2021 `nfrobe_receiver` y se generarán registros NetFlow versión 5 `nfprobe_version`.

Se activa el plugin `memory` para comprobar que los registros generados por el agente son los mismos que se representan en el colector.

CAPÍTULO 4. Pruebas del sistema

En este capítulo se van a presentar la serie de pruebas que se han realizado para comprender el funcionamiento del software `pmacct`. Describiremos los resultados obtenidos resaltando las deficiencias que se puedan presentar.

4.1 Puesta en marcha del daemon BGP

En esta prueba vamos a verificar el funcionamiento del daemon BGP activado en la máquina Colector. Vamos a describir la información que se presenta y si es la información esperada.

Activamos el daemon `nfacctd` en máquina colector, donde se ha establecido la configuración del daemon BGP ya presentada en la Fig. 3.14. La información que se recibe es la siguiente (Figura 4.1- Figura 4.2):

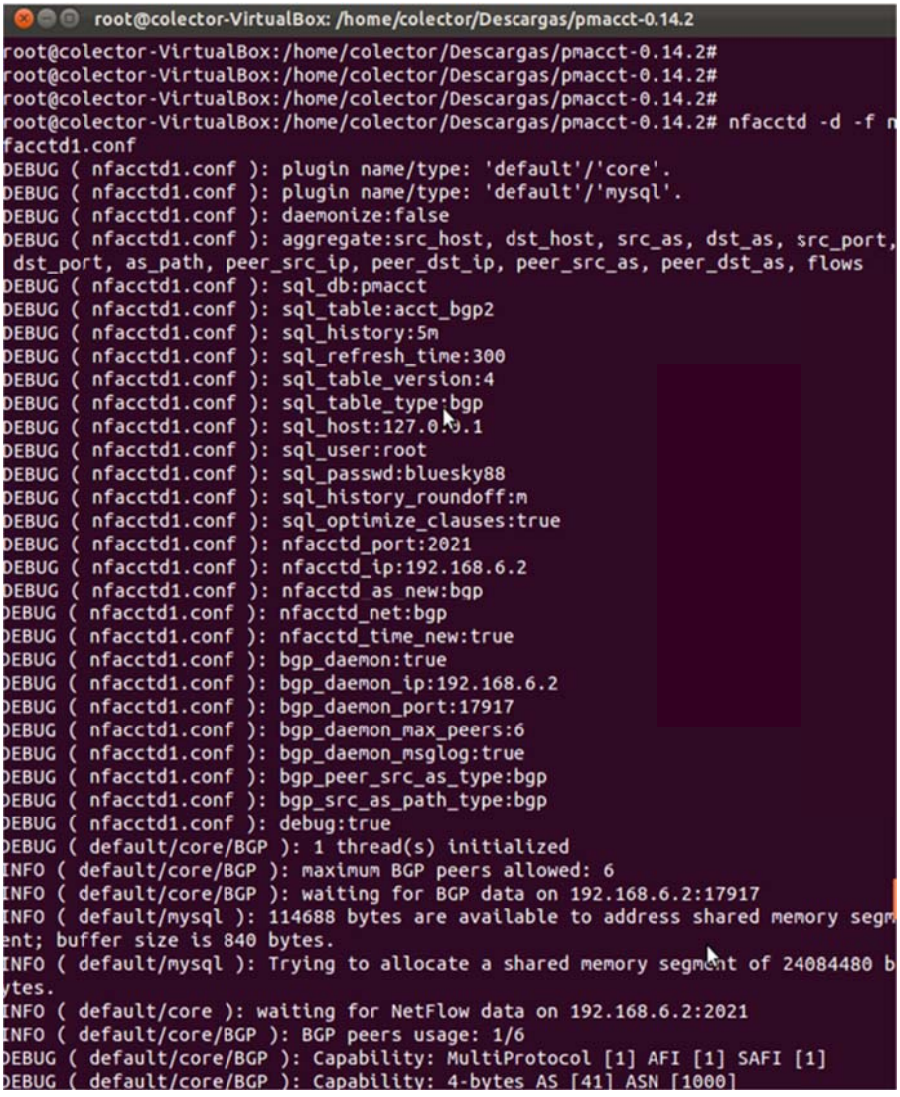


Fig. 4.1 Activación del daemon BGP en la máquina Colector

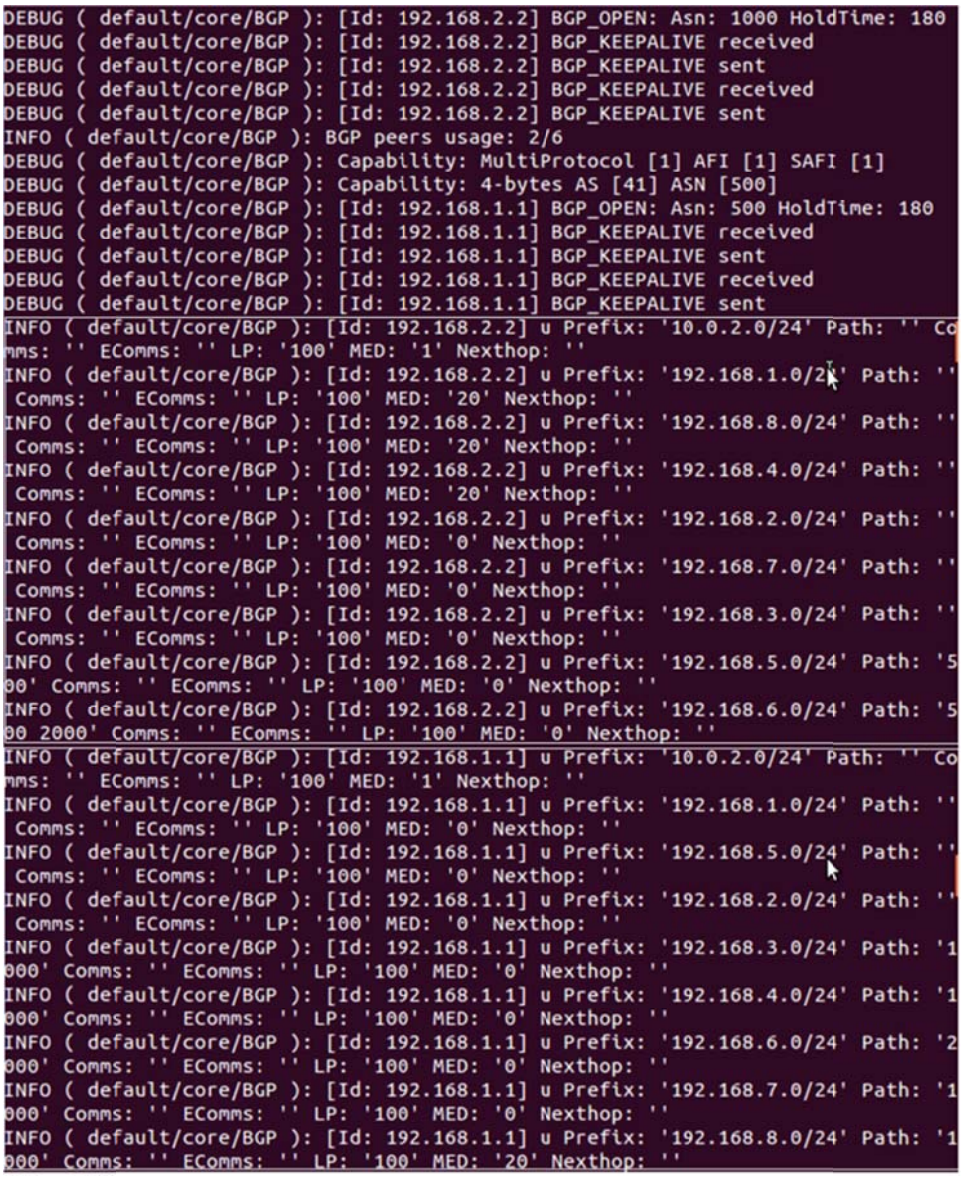


Fig. 4.2 Recepción de las tablas de encaminamiento

La información presentada en la figura es el establecimiento de conexión con 2 vecinos BGP `BGP peers usage:1/6 ; BGP peers usage:2/6`. Como ya se comentó en el apartado 3.3 el daemon BGP que incorpora `pmacct` va a actuar de forma pasiva, es decir, esperará a conexiones entrantes `BGP_OPEN: ASN: xx Holdtime: 180`. Éste responderá de su existencia con un mensaje `BGP_KEEPALIVE received`. Una vez establecida la conexión con el dispositivo vecino se recibirá la tabla de encaminamiento del mismo, indicando los prefijos de redes que él conoce y el camino que debe seguir para llegar a ellos `INFO (default/core/BGP): [Id:192.168.2.2] u Prefix: '192.168.6.0/24' Path: '500 2000'`.

Comprobamos que el establecimiento de conexión se ha realizado correctamente consultando en el dispositivo R1 `sh ip bgp summary` (Figura 4.3):

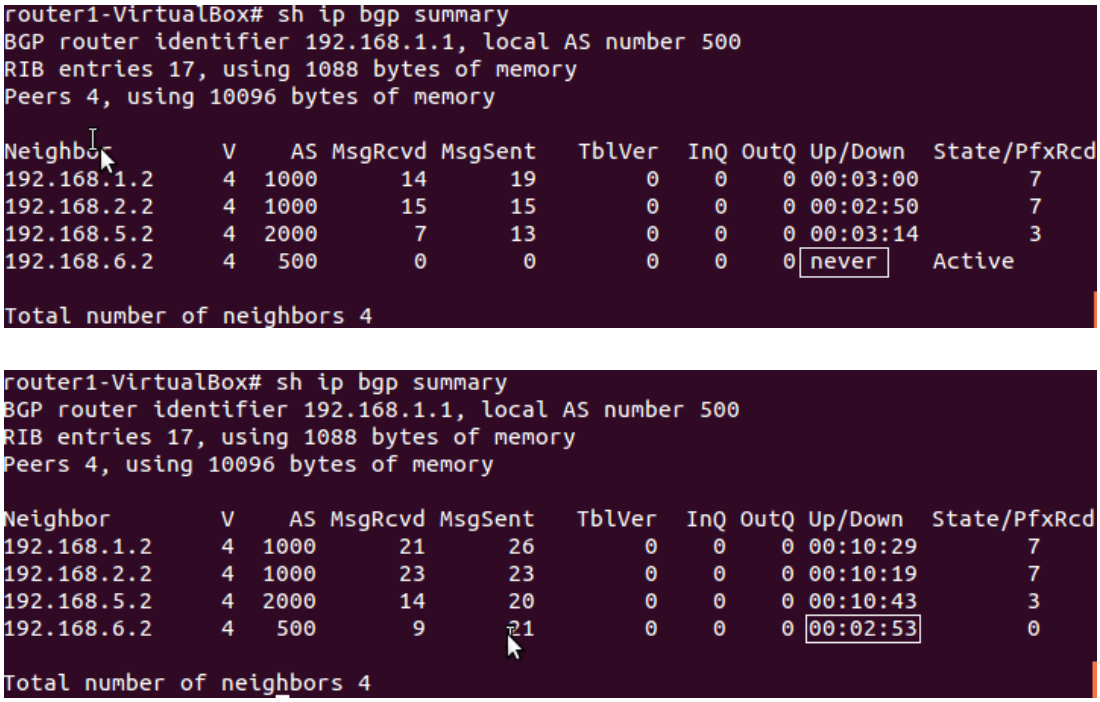


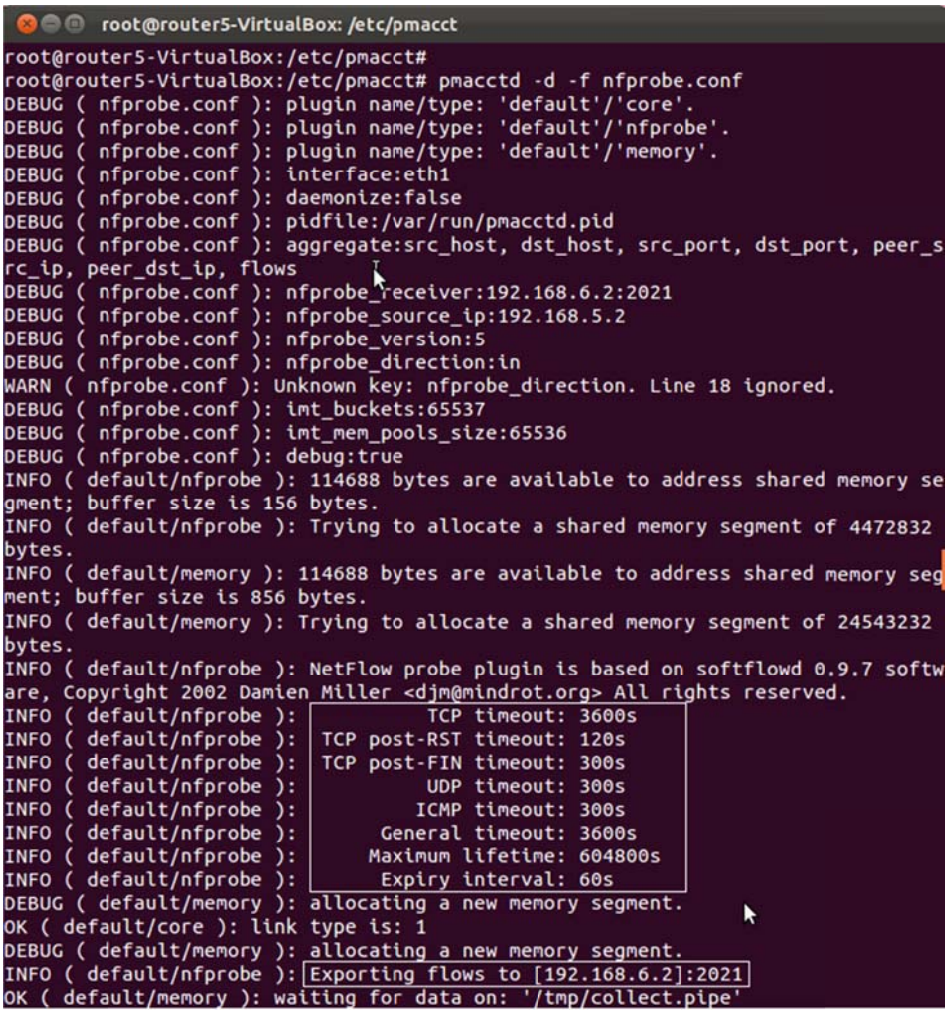
Fig. 4.3 Resumen de conexiones BGP en R1

En la primera de ellas antes de activar el daemon BGP se puede comprobar que no existe relación de vecindad con la máquina Colector `192.168.6.2`; Una vez activado el daemon BGP se demuestra que existe una relación de vecindad. Como se documenta en el apartado 3.3.1 una de las deficiencias que presenta el daemon BGP proporcionado por `pmacct` es que solo entiende de conexiones IGP, es decir, para establecer la relación de vecindad se ha de indicar que el AS al que pertenece es el mismo que el de la máquina vecina AS `500` como ya se demuestra en la figura 2.3.

4.2 Recepción de datos plugin nfprobe

En esta prueba vamos a verificar que los registros generados por el plugin nfprobe proporcionado por `pmacct` activado en la máquina R5 se corresponden con los datos que presenta el colector NetFlow ubicado en Colector y representado en la base de datos `pmacct`. Se van a hacer pruebas generando tráfico desde máquina R1 enviando un ping y el envío de un vídeo desde la máquina R3.

Activamos el plugin `nfprobe` en la máquina R5. La información que se recibe es la siguiente (Figura 4.4):



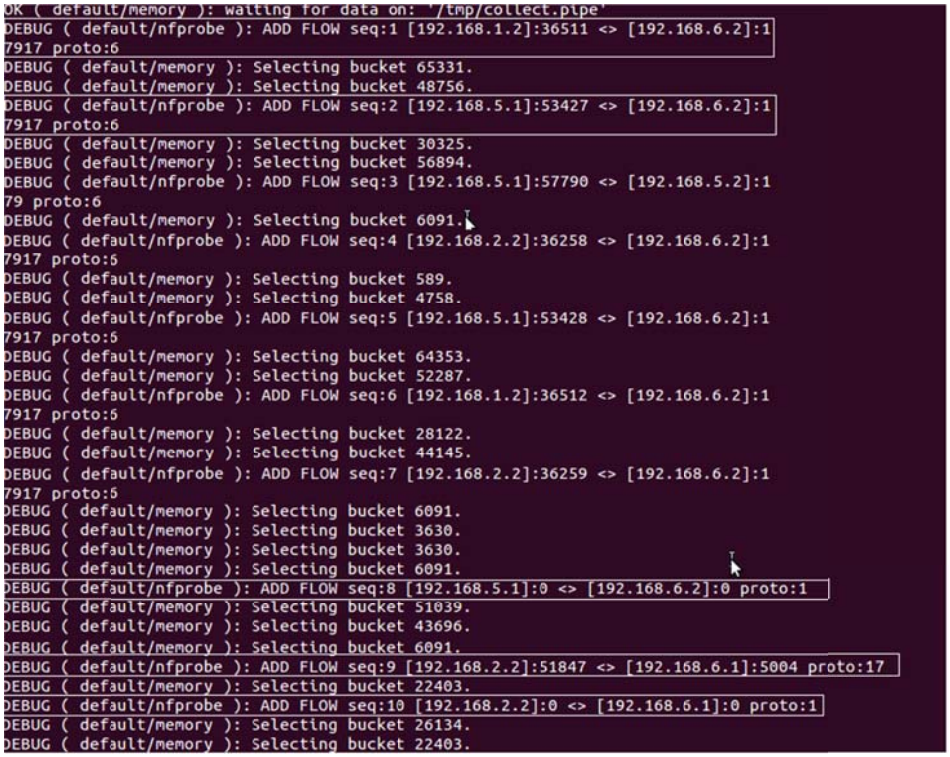


Fig. 4.4 Flujos generados por el plugin nfprobe en la máquina R5

La información presentada en la figura es la activación del plugin `nfprobe` donde se informan los campos configurados como: definir el colector de registros netflow `Exporting flows to [192.168.6.2]:2021`, definir la versión de los registros NetFlow `nfprobe_version:5` o definir los parámetros para determinar el fin de flujo `TCP timeout:3600s UDP timeout: 300s` como ya se presentó en la figura 1.8. Se verifica la creación de los registros `ADD FLOW seq:x [Dirección IP origen]:puerto <> [Dirección IP destino]puerto.` Los primeros flujos reflejan el intercambio de mensajes `BGP_Keepalives` entre R2 – Colector / R1 – Colector. Hemos destacado también la creación del registro al enviar un ping desde la máquina R1 a máquina Colector `ADD FLOW seq:8 [192.168.5.1]0 <> [192.168.6.2]0` y la transmisión de un vídeo desde máquina R3 a máquina R5 `ADD FLOW seq:9 [192.168.2.2]51847 <> [192.168.6.1]:5004.`

A continuación analizamos los datos recibidos en el colector y comprobamos si concuerdan con los registros generados en la máquina R5. También se va a verificar si funciona correctamente el envío de datos al sistema de almacenamiento de datos MySQL (Figura 4.5 – Figura 4.6):

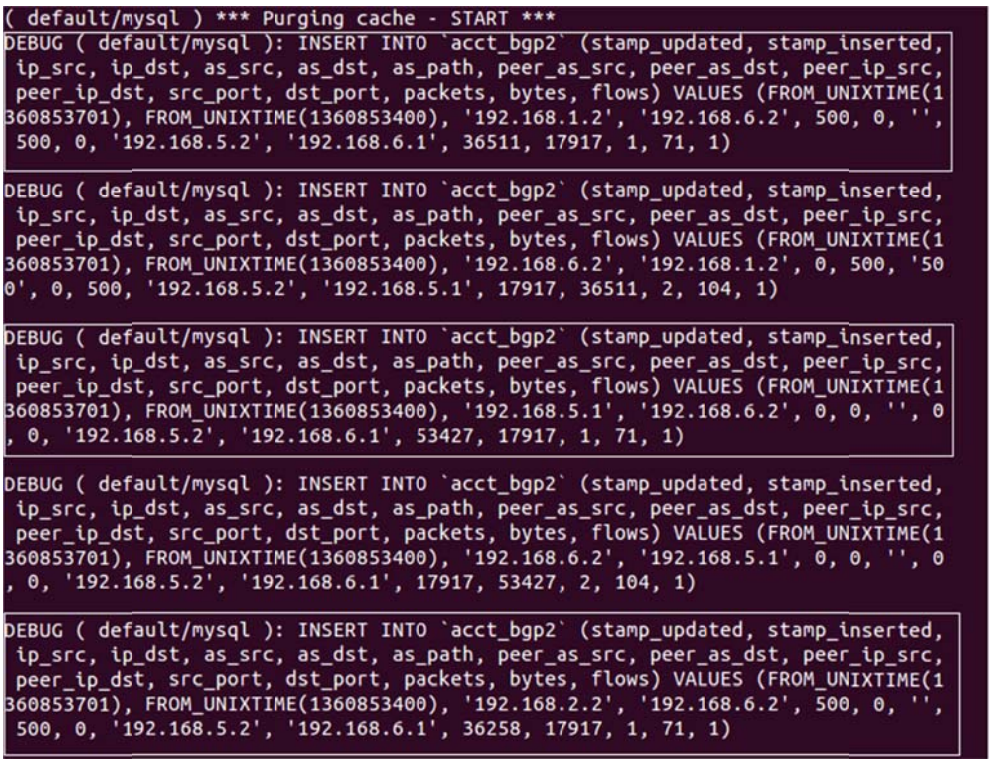
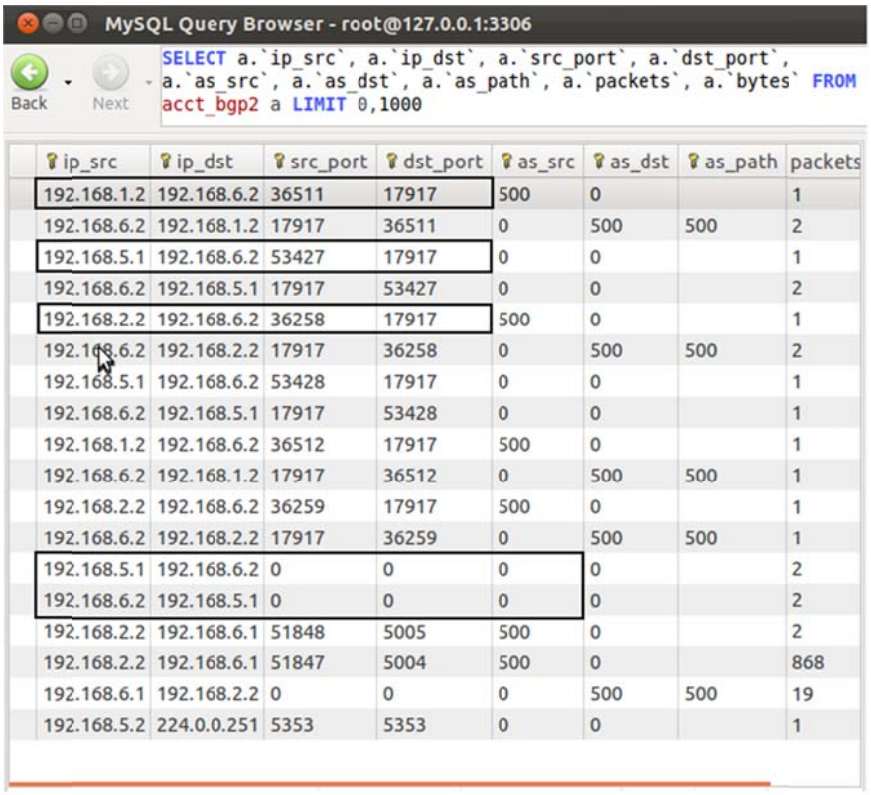


Fig. 4.5 Recepción de registros NetFlow en la máquina Colector



ip_src	ip_dst	src_port	dst_port	as_src	as_dst	as_path	packets
192.168.1.2	192.168.6.2	36511	17917	500	0		1
192.168.6.2	192.168.1.2	17917	36511	0	500	500	2
192.168.5.1	192.168.6.2	53427	17917	0	0		1
192.168.6.2	192.168.5.1	17917	53427	0	0		2
192.168.2.2	192.168.6.2	36258	17917	500	0		1
192.168.6.2	192.168.2.2	17917	36258	0	500	500	2
192.168.5.1	192.168.6.2	53428	17917	0	0		1
192.168.6.2	192.168.5.1	17917	53428	0	0		1
192.168.1.2	192.168.6.2	36512	17917	500	0		1
192.168.6.2	192.168.1.2	17917	36512	0	500	500	1
192.168.2.2	192.168.6.2	36259	17917	500	0		1
192.168.6.2	192.168.2.2	17917	36259	0	500	500	1
192.168.5.1	192.168.6.2	0	0	0	0		2
192.168.6.2	192.168.5.1	0	0	0	0		2
192.168.2.2	192.168.6.1	51848	5005	500	0		2
192.168.2.2	192.168.6.1	51847	5004	500	0		868
192.168.6.1	192.168.2.2	0	0	0	500	500	19
192.168.5.2	224.0.0.251	5353	5353	0	0		1

Fig.4.6 Envío de registros NetFlow a la base de datos MySQL

Comprobamos que los datos registrados concuerdan con los datos generados por el agente NetFlow ubicado en la máquina R5 que se presentan en la Fig. 4.3. El primer registro guardado corresponde al primer flujo añadido `ip_src:192.168.1.2;ip_dst:192.168.6.2:17917;src_port:36511; dst_port:17917.` Los demás campos que aparecen registrados como `as_src, as_dst` … son valores que el agente NetFlow no genera pero que se registran por la activación del daemon BGP ya presentado en el anterior apartado.

Se verifica el correcto funcionamiento entre el software `pmacct` y la base de datos `pmacct`. Los datos presentes en la primera parte de la figura corresponden a los presentados en la base de datos.

Se encuentra una deficiencia en cuanto al funcionamiento del plugin nfprobe en concreto en la directiva `nfprobe_direction:in.` El funcionamiento de ésta debería ser que solo se capta el tráfico de entrada en la interfaz donde está escuchando el agente NetFlow. En la figura podemos demostrar que esto no es así y que también se registran los flujos de salida como el segundo registro guardado en MySQL `ip_src:192.168.6.2;ip_dst:192.168.1.2; src_port:17917;dst_port:36511.`Los registros no se presentan por pantalla pero se ha comprobado mediante Wireshark que sí son enviados.

4.3 Diferencias entre plugin nfprobe y aplicación fprobe

Como se presentó en el apartado 2.1.1.5 se han utilizado dos aplicaciones para simular el comportamiento de un agente NetFlow. Por una parte utilizamos la aplicación presentada por `pmacct` activada mediante el plugin `nfprobe` en aquellas máquinas donde se ha instalado el software `pmacct` que son Colector, R5, R1 y R2. Por otra parte, en aquellas máquinas donde no se despliega `pmacct` se ha utilizado la aplicación `fprobe`. En esta prueba vamos a comprobar que diferencias existen entre los dos agentes NetFlow. Para ello se van a consultar los registros generados por el agente ubicado en R5 y que son enviados al colector NetFlow instalado en la máquina Colector. Se van a comparar con los registros generados por el agente instalado en la máquina R1 y que son enviados al colector NetFlow instalado en la propia máquina.

La prueba va a consistir en enviar un flujo de datos desde la máquina R4 con `ip_src:192.168.4.2` a la máquina R5 `ip_dst:192.168.6.1`.Los registros NetFlow se van a consultar utilizando el software `wireshark`, ya que de esta manera nos vamos a asegurar de comparar estrictamente los datos generados por el agente y no comparar los datos enviados a la base de datos donde entre otros se ha añadido información por el uso del daemon BGP.

A continuación se van a mostrar las capturas tomadas en `wireshark` (Figura 4.7 – Figura 4.8):

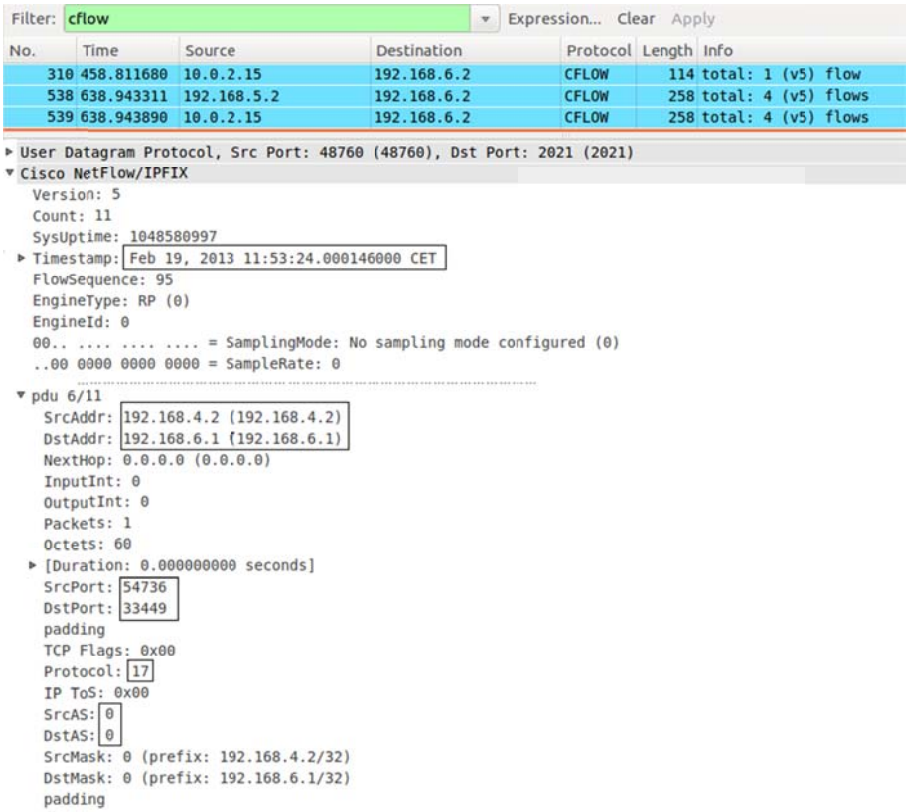


Fig. 4.7 Recepción de registro NetFlow en la máquina Colector con software Wireshark

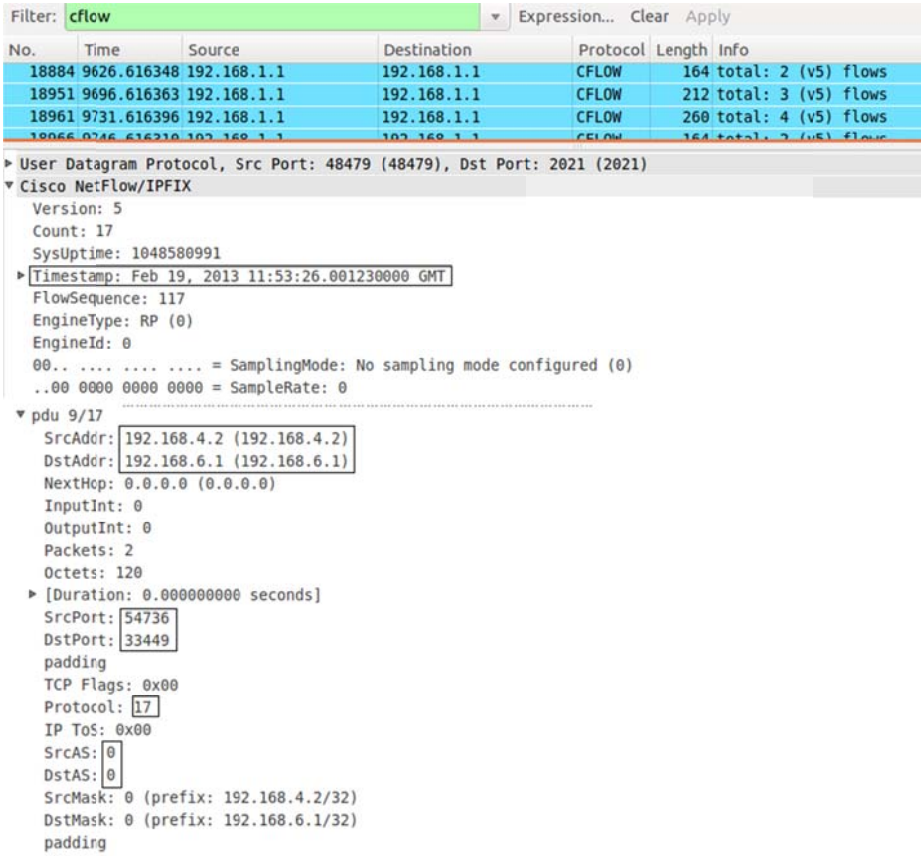


Fig. 4.8 Recepción de registro NetFlow en la máquina R1con software Wireshark

En la captura de la izquierda mostramos el registro recibido en la máquina Colector. Destacamos la hora en la que fue recibido el registro `Timestamp: Feb 19, 2013 11:53:24`. Se informa de un flujo de datos UDP `Protocol:17` generado por la máquina R2 `SrcAddr: (192.168.4.2)`y con destino la máquina R5 `DstAddr: (192.168.6.1)`con puerto origen y destino `SrcPort: 54736 / DstPort: 33449.`

En la captura de la derecha se muestra el registro recibido en la máquina R1. Se muestra que es un registro generado en el mismo instante que en la máquina Colector `Timestamp: Feb 19, 2013 11:53:26`. Se comprueba que los datos tanto generados por el agente NetFlow que incluye `pmacct` como los generados por aplicación `fprobe` contienen la misma información. También se corrobora la información expuesta en el apartado 3.3, dónde se describe una deficiencia en los dos agentes NetFlow, que es la no obtención de valores referentes a encaminamiento como `SrcAS: 0 / DstAS: 0`.

Destacamos que las principales diferencias presentes entre los agentes NetFlow han sido encontradas en las directivas de configuración de la aplicación `fprobe`. Como se muestra en la Fig.1.8 el agente NetFlow que presenta `pmacct` hace distinción a la hora de determinar el fin de flujo dependiendo del protocolo de transporte utilizado UDP/TCP, mientras que la aplicación `fprobe` solo va a determinar un valor de fin de flujo para todos los

registros. Por otra parte, se ha detectado que la aplicación `fprobe` no presenta ninguna directiva de configuración para discriminar entre tráfico de entrada o salida. Ésta diferencia no es relevante, ya que como se demostró en el apartado 4.2, el funcionamiento de la directiva `nfprobe_direction` presente en el agente NetFlow de `pmacct` no es correcto.

4.4 Diferencias entre la recepción de datos del script networks.txt y daemon BGP

Como se presentó en el apartado 3.3, `pmacct` presenta una serie de soluciones de cara a resolver la deficiencia presentada por los agentes NetFlow sobre la obtención de valores referentes a encaminamiento. En esta prueba vamos a comprobar que diferencias existen entre las dos soluciones propuestas por `pmacct` que son la obtención de valores referentes a encaminamiento como `as_src`, `as_dst`, `peer_as_src`, `peer_as_dst` o `as_path` activando el daemon BGP o con la utilización del fichero networks.txt.

La prueba va a consistir en comparar los registros guardados en las bases de datos `pmacct` de la máquina Colector, en este caso la tabla `acct_bgp2` y de la máquina R1 siendo la tabla donde se guardan los registros `acct_bgp1`.

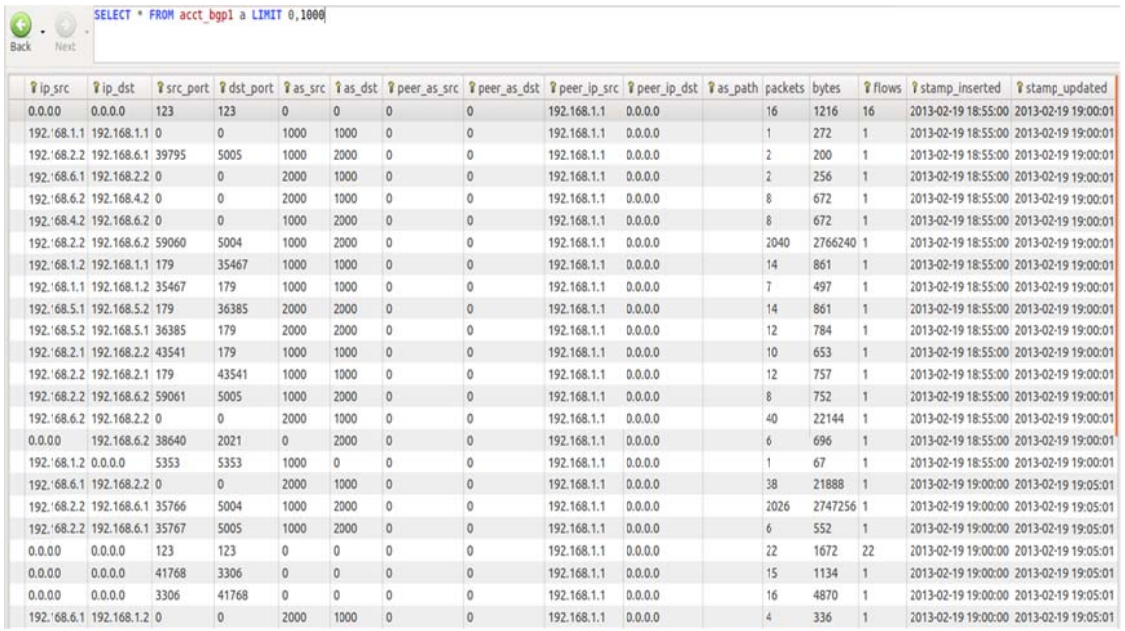
En la máquina Colector se van observar que valores referentes a encaminamiento se van a generar con la utilización del daemon BGP (Figura 4.9) mientras que en la máquina R5 van a ser generados por el fichero networks.txt (Figura 4.10).

A continuación se van a presentar las capturas tomas y analizar sus diferencias:

```
SELECT * FROM acct_bgp2 a LIMIT 0,1000
```

ip_src	ip_dst	src_port	dst_port	as_src	as_dst	peer_as_src	peer_as_dst	peer_ip_src	peer_ip_dst	as_path	packets	bytes	flows	stamp_inserted	stamp_updated
192.168.5.2	192.168.4.2	47524	33448	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	54235	33445	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	47239	33436	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	57957	33446	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	59872	33440	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	37121	33441	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	42170	33443	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	56478	33437	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	33939	33442	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	46458	33441	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.5.2	192.168.4.2	40861	33438	0	1000	0	500	192.168.6.1	192.168.5.1	500 1000	1	60	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.2.2	192.168.6.2	38529	5004	500	0	500	0	192.168.6.1	192.168.6.1		928	1258368	1	2013-02-20 04:25:00	2013-02-20 04:30:01
192.168.2.2	192.168.6.2	54865	179	500	0	500	0	192.168.6.1	192.168.6.1		1	60	1	2013-02-20 04:30:00	2013-02-20 04:35:01
192.168.5.2	192.168.2.2	179	54865	0	500	0	500	192.168.6.1	192.168.5.1	500	1	40	1	2013-02-20 04:30:00	2013-02-20 04:35:01
192.168.5.1	192.168.6.2	49542	5005	0	0	0	0	192.168.6.1	192.168.6.1		2	184	1	2013-02-20 04:30:00	2013-02-20 04:35:01
192.168.5.2	192.168.5.1	0	0	0	0	0	0	192.168.6.1	192.168.6.1		17	9336	1	2013-02-20 04:30:00	2013-02-20 04:35:01
192.168.5.1	192.168.6.2	49541	5004	0	0	0	0	192.168.6.1	192.168.6.1		812	1101072	1	2013-02-20 04:30:00	2013-02-20 04:35:01
192.168.5.1	192.168.5.2	179	36484	0	0	0	0	192.168.6.1	192.168.6.1		14	861	1	2013-02-20 04:30:00	2013-02-20 04:35:01

Fig. 4.9 Registros guardados en la base de datos pmacct en la máquina Colector



```
SELECT * FROM acct_bgp1 a LIMIT 0,1000
```

Back Next

ip_src	ip_dst	src_port	dst_port	as_src	as_dst	peer_as_src	peer_as_dst	peer_ip_src	peer_ip_dst	as_path	packets	bytes	flows	stamp_inserted	stamp_updated
0.0.0.0	0.0.0.0	123	123	0	0	0	0	192.168.1.1	0.0.0.0		16	1216	16	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.1.1	192.168.1.1	0	0	1000	1000	0	0	192.168.1.1	0.0.0.0		1	272	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.2.2	192.168.6.1	39795	5005	1000	2000	0	0	192.168.1.1	0.0.0.0		2	200	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.6.1	192.168.2.2	0	0	2000	1000	0	0	192.168.1.1	0.0.0.0		2	256	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.6.2	192.168.4.2	0	0	2000	1000	0	0	192.168.1.1	0.0.0.0		8	672	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.4.2	192.168.6.2	0	0	1000	2000	0	0	192.168.1.1	0.0.0.0		8	672	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.2.2	192.168.6.2	59060	5004	1000	2000	0	0	192.168.1.1	0.0.0.0		2040	2766240	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.1.2	192.168.1.1	179	35467	1000	1000	0	0	192.168.1.1	0.0.0.0		14	861	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.1.1	192.168.1.2	35467	179	1000	1000	0	0	192.168.1.1	0.0.0.0		7	497	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.5.1	192.168.5.2	179	36385	2000	2000	0	0	192.168.1.1	0.0.0.0		14	861	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.5.2	192.168.5.1	36385	179	2000	2000	0	0	192.168.1.1	0.0.0.0		12	784	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.2.1	192.168.2.2	43541	179	1000	1000	0	0	192.168.1.1	0.0.0.0		10	653	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.2.2	192.168.2.1	179	43541	1000	1000	0	0	192.168.1.1	0.0.0.0		12	757	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.2.2	192.168.6.2	59061	5005	1000	2000	0	0	192.168.1.1	0.0.0.0		8	752	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.6.2	192.168.2.2	0	0	2000	1000	0	0	192.168.1.1	0.0.0.0		40	22144	1	2013-02-19 18:55:00	2013-02-19 19:00:01
0.0.0.0	192.168.6.2	38640	2021	0	2000	0	0	192.168.1.1	0.0.0.0		6	696	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.1.2	0.0.0.0	5353	5353	1000	0	0	0	192.168.1.1	0.0.0.0		1	67	1	2013-02-19 18:55:00	2013-02-19 19:00:01
192.'68.6.1	192.168.2.2	0	0	2000	1000	0	0	192.168.1.1	0.0.0.0		38	21888	1	2013-02-19 19:00:00	2013-02-19 19:05:01
192.'68.2.2	192.168.6.1	35766	5004	1000	2000	0	0	192.168.1.1	0.0.0.0		2026	2747256	1	2013-02-19 19:00:00	2013-02-19 19:05:01
192.'68.2.2	192.168.6.1	35767	5005	1000	2000	0	0	192.168.1.1	0.0.0.0		6	552	1	2013-02-19 19:00:00	2013-02-19 19:05:01
0.0.0.0	0.0.0.0	123	123	0	0	0	0	192.168.1.1	0.0.0.0		22	1672	22	2013-02-19 19:00:00	2013-02-19 19:05:01
0.0.0.0	0.0.0.0	41768	3306	0	0	0	0	192.168.1.1	0.0.0.0		15	1134	1	2013-02-19 19:00:00	2013-02-19 19:05:01
0.0.0.0	0.0.0.0	3306	41768	0	0	0	0	192.168.1.1	0.0.0.0		16	4870	1	2013-02-19 19:00:00	2013-02-19 19:05:01
192.'68.6.1	192.168.1.2	0	0	2000	1000	0	0	192.168.1.1	0.0.0.0		4	336	1	2013-02-19 19:00:00	2013-02-19 19:05:01

Fig. 4.10 Registros guardados en la base de datos pmacct en la máquina R5

La primera diferencia la encontramos en los campos `as_src / as_dst`. Observamos que para el caso de utilizar el fichero networks.txt como se presenta en la Fig. 4.5, este campo siempre va a registrar cual es el AS origen de los datos y el AS destino de los mismos. En el caso de utilizar el daemon BGP se observa que para un rango de `ip_src / ip_dst` estos campos serán marcados como `0`. La explicación a esta forma de actuar deriva a que la máquina donde está ubicado el agente NetFlow, en este caso la máquina R5, todos los flujos de datos que se generen o tengan como destino una máquina adyacente son interpretados como una IP local. Para el caso de la máquina R5, todo tráfico que tenga una IP origen o IP destino en el rango de direcciones `192.168.6.0/24 / 192.168.5.0/24` serán marcados como `0`.

La segunda diferencia la encontramos en los campos `peer_as_src / peer_as_dst / as_path`. Se observa que utilizando el fichero networks.txt, esta información no va a poder ser interpretada y por lo tanto va a ser marcada como `0`. En el caso de utilizar el daemon BGP, para la obtención de datos referentes a `peer_as_src / peer_as_dst` nos encontramos con la misma actuación que para los campos `as_src / as_dst`, aquellos datos con origen o destino a una máquina adyacente a R5 serán marcados como `0`. En la obtención de datos referentes al campo `as_path` este será marcada siempre y cuando el tráfico sea saliente. Si el tráfico es de entrada el campo `as_path` será marcado como `0`. Ésto tiene sentido ya que quien genera los informes, en este caso el agente instalado en la máquina R5, va a saber el camino que deben seguir los paquetes que ha encaminado, pero no va a obtener la información del camino que han seguido los paquetes hasta llegar a él.

CAPÍTULO 5. ANÁLISIS DE LOS DATOS

En este capítulo se va a documentar como tratar los registros guardados en las bases de datos y que consultas son necesarias realizar para obtener las matrices de tráfico. Se van a diferenciar tres casos; obtención de matrices de tráfico a nivel de AS, obtención de matrices de tráfico a nivel de router y para finalizar obtención de matrices de tráfico a nivel de interfaz.

Una matriz de tráfico está definida por tres campos; origen de los datos, destino de los datos en un espacio de tiempo determinado.

5.1 Obtención de la matriz de tráfico a nivel de AS

El primer ejemplo que se va a mostrar es la obtención de la matriz de tráfico con:

- Origen de los datos: AS2000
- Destino de los datos: AS1000
- Tiempo: 2013-02-20 04:25:00 – 2013-02-20 04:30:01

Se van a describir dos maneras de obtener la matriz de tráfico; la primera de ellas haciendo referencia al campo `as_src` (Figura 5.1) y la segunda realizando la consulta asociando las `ip_src` a un AS determinado (Figura 5.2):

```
mysql> select as_src, as_dst, count(*), sum(bytes), stamp_inserted, stamp_update
d from acct_bgp2 where as_src = '0' and as_dst = '1000' and stamp_inserted = "20
13-02-20 04:25:00" group by as_src,as_dst;
+--------+--------+----------+-----------+---------------------+--------------
------+
| as_src | as_dst | count(*) | sum(bytes) | stamp_inserted      | stamp_updated
      |
+--------+--------+----------+-----------+---------------------+--------------
------+
|      0 |   1000 |      104 |       6264 | 2013-02-20 04:25:00 | 2013-02-20 04:
30:01 |
+--------+--------+----------+-----------+---------------------+--------------
------+
1 row in set (0.00 sec)
```

Fig. 5.1 Consulta MySQL para obtener la TM entre AS2000-AS1000

Como se presentó en el apartado anterior, los datos registrados por el agente NetFlow sobre el tráfico generado por routers adyacentes va a ser marcado como `as_src: 0/ as_dst:0`. Por ello la consulta mysql, para la obtención de registros generados en el AS2000 (AS donde se encuentra el agente NetFlow) con destino el AS1000 se realiza de la siguiente manera `where as_src ='0' and as_dst= '1000'`. La información que se presenta en la figura, es la suma total del tráfico `sum(bytes)` entre ASs.

La segunda manera de obtener la matriz de tráfico es asociando la dirección IP origen de los routers pertenecientes al AS2000, en este caso, `ip_src:'192.168.6.2',ip_src:'192.168.5.2'`. con el AS1000.

```
mysql> select ip_src, as_dst, count(*), sum(bytes), stamp_inserted, stamp_update
d from acct_bgp2 where ip_src in ('192.168.5.2','192.168.6.2') and as_dst = '100
0' and stamp_inserted = "2013-02-20 04:25:00" group by ip_src,as_dst;
+-------------+--------+----------+-----------+---------------------+----------
-----------+
| ip_src      | as_dst | count(*) | sum(bytes) | stamp_inserted      | stamp_upd
ated       |
+-------------+--------+----------+-----------+---------------------+----------
-----------+
| 192.168.5.2 |   1000 |       16 |        960 | 2013-02-20 04:25:00 | 2013-02-2
0 04:30:01 |
| 192.168.6.2 |   1000 |       88 |       5304 | 2013-02-20 04:25:00 | 2013-02-2
0 04:30:01 |
+-------------+--------+----------+-----------+---------------------+----------
-----------+
2 rows in set (0.00 sec)
```

Fig.5.2 Consulta MySQL para obtener la TM entre AS2000-AS1000

Se verifica que la información obtenida es la misma realizando la suma del campo `sum(bytes)`donde se obtiene un resultado de 6264 que es el mismo que se presenta en la Fig.5.1. La información obtenida con esta consulta es más detallada que la anterior, ya que, podemos diferenciar, la suma total de bytes enviados por un router u otro.

5.2 Obtención de la matriz de tráfico a nivel de router

En este apartado vamos a mostrar que consulta se ha de realizar en el software MySQL para obtener la matriz de tráfico a nivel de router, entre la máquina Colector y la máquina R4 (Figura 5.3).

Para obtener esta matriz de tráfico se ha de asociar entre la/s direcciónes IP desde donde se generan los datos con la/s direcciones IP a donde se envían los datos. En este caso será `ip_src:'192.168.6.2'/ ip_dst:'192.168.4.2','192.168.7.2',192.168.8.1'`.

```
mysql> select ip_src, ip_dst, count(*), sum(bytes), stamp_inserted, stamp_update
d from acct_bgp2 where ip_src = '192.168.6.2' and ip_dst in ('192.168.4.2','192.
168.7.2','192.168.8.1') and stamp_inserted = "2013-02-20 04:25:00" group by ip_s
rc,ip_dst;
+-------------+-------------+----------+-----------+---------------------+-----
----------------+
| ip_src      | ip_dst      | count(*) | sum(bytes) | stamp_inserted      | stam
p_updated       |
+-------------+-------------+----------+-----------+---------------------+-----
----------------+
| 192.168.6.2 | 192.168.4.2 |       87 |       5220 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.6.2 | 192.168.7.2 |        1 |         84 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
+-------------+-------------+----------+-----------+---------------------+-----
----------------+
2 rows in set (0.00 sec)
```

Fig. 5.3 Consulta MySQL para obtener la TM entre Colector y R4

La información obtenida realizando esta consulta, es la suma total del tráfico `sum(bytes)` enviado desde la máquina Colector `ip_src:'192.168.6.2'` con destino la máquina R4, en concreto `ip_dst:'192.168.4.2', '192.168.7.2'` en el espacio de tiempo `stamp_inserted:2013-02-20 04:25:00 / stamp_updated:2013-02-20 04:30:01`. No se presenta ningún registro asociado a la dirección IP `192.168.8.1`, ya que en el espacio de tiempo consultado no se ha generado ningún envío de datos.

5.3 Obtención de la matriz de tráfico a nivel de interfaz

En este apartado vamos a mostrar que consulta se ha de realizar en la base de datos MySQL, para obtener la matriz de tráfico a nivel de interfaz de red, diferenciando entre IP origen e IP destino (Figura 5.4):

```
mysql>
mysql> select ip_src, ip_dst, count(*), sum(bytes), stamp_inserted, stamp_update
d from acct_bgp2 where stamp_inserted = "2013-02-20 04:25:00" group by ip_src,ip
_dst;
+-------------+-------------+----------+------------+---------------------+-----
----------------+
| ip_src      | ip_dst      | count(*) | sum(bytes) | stamp_inserted      | stam
p_updated       |
+-------------+-------------+----------+------------+---------------------+-----
----------------+
| 192.168.1.2 | 192.168.5.2 |        1 |        264 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.1.2 | 192.168.6.2 |        1 |        264 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.2.2 | 192.168.6.2 |        4 |    1258672 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.4.2 | 192.168.5.2 |        1 |        528 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.5.1 | 192.168.5.2 |        1 |        264 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.5.1 | 192.168.6.2 |        1 |        264 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.5.1 | 224.0.0.251 |        1 |        169 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.5.2 | 192.168.4.2 |       16 |        960 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.5.2 | 224.0.0.251 |        1 |        490 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.6.2 | 192.168.2.2 |        3 |       9992 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.6.2 | 192.168.4.2 |       87 |       5220 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
| 192.168.6.2 | 192.168.7.2 |        1 |         84 | 2013-02-20 04:25:00 | 2013
-02-20 04:30:01 |
+-------------+-------------+----------+------------+---------------------+-----
----------------+
12 rows in set (0.00 sec)
```

Fig.5.4 Consulta MySQL para obtener la TM a nivel de interfaz

La información obtenida con esta consulta, es la suma del tráfico total generado, en el espacio de tiempo `stamp_inserted:2013-02-20 04:25:00 / stamp_updated:2013-02-20 04:30:01` diferenciando entre el origen de los datos `ip_src` y el destino de los mismos `ip_dst`.

CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS

Con la realización de este proyecto se ha podido estudiar y analizar el funcionamiento de una aplicación para el control de tareas relacionadas con la monitorización del tráfico. Se ha diseñado un escenario de red intentando aproximarse lo más posible a la realidad, sobre una sola máquina física mediante máquinas virtuales, donde se han configurado protocolos de encaminamiento interno (OSPF) y externo (BGP) como el protocolo para gestionar el tráfico de red NetFlow.

Se han presentado dos alternativas de software para generar informes de red. Por un lado se ha presentado el agente NetFlow que incorpora `pmacct` y se ha comparado por un agente NetFlow que podemos descargar de la red. Se ha concluido que las dos alternativas son validas, ya que los registros obtenidos contienen la misma información, pero nos decantamos por el agente presentado por `pmacct` ya que ofrece más alternativas de configuración como diferenciar entre flujos transportados por un protocolo de transporte u otro. Ambas alternativas presentan una deficiencia común en cuanto a la generación de información referente a campos de encaminamiento.

Se han presentado dos alternativas para registrar parámetros referentes a encaminamiento como solución a la deficiencia de los agentes NetFlow, presentados por el software `pmacct`. Por un lado se presenta una solución completa como es la activación del daemon BGP que actuará de manera conjunta con un software externo como es `quagga`. La segunda solución propuesta es la obtención de parámetros a través de una información presentada en un fichero generado manualmente. Concluimos que la solución más robusta es la activación del daemon BGP, no exenta de deficiencias como solo ofrecer la posibilidad de trabajar como un protocolo de encaminamiento interior, ya que permite la obtención de más información referente a encaminamiento.

Se ha presentado una solución para guardar los registros generados como es MySQL. Hemos determinado que ha sido una buena solución por diferentes motivos. Uno de ellos es que es un software muy extendido, por lo que podemos encontrar mucha documentación referente al mismo. Esto ha facilitado en gran medida el realizar las diferentes consultas para obtener las matrices de tráfico. Otra ventaja que nos ofrece, es la de ofrecer una herramienta que nos permite trabajar con los datos con una mejor visualización que la dada por la Shell.

La conclusión final es que `pmacct` es un software robusto para trabajar temas relacionados con la monitorización de tráfico, ya que los resultados obtenidos se adecuan con la realidad. Es una herramienta que presenta soluciones a deficiencias que se han encontrado en el mismo y está en continua evolución ofreciendo nuevas alternativas de estudio como puede ser desplegarlo en redes móviles. Otra ventaja que ofrece es la de poder trabajar con aplicaciones

externas como en nuestro caso ha sido `quagga`, `fprobe` o `MySQl`.

Como líneas futuras, para corroborar la robustez del software `pmacct` sería importante desplegarlo en una red real, con equipamiento como un router Cisco que entienda NetFlow dando soporte a una red de ordenadores. Con esto podremos comparar el funcionamiento de un agente NetFlow configurado en un equipo real y si los registros referentes a parámetros de encaminamiento concuerdan con los obtenidos con las soluciones aquí estudiadas. Otra ventaja que nos ofrecería trabajar sobre un escenario real, sería solucionar el problema con el que nos hemos encontrado a la hora de generar grandes flujos de tráfico. Con esto se podrán estudiar los diferentes filtros que presentan tanto el software MySQL como el agente NetFlow en términos de número de bytes enviados, o número de paquetes enviados, entre otros.

GLOSARIO

AS	Autonomous System
BGP	Border Gateway Protocol
CPD	Centro de Procesado de Datos
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
EGP	External Gateway Protocol
IGP	Internal Gateway Protocol
IP	Internet Protocol
ISP	Internet Service Provider
IXP	Internet eXchange Point
MIB	Management Information Base
NAT	Network Address Translation
nfacctd	Netflow accounting daemon
OSPF	Open Shortest Path First
pmacct	Promiscous mode IP accounting package
pmacctd	Promiscuous mode accounting daemon
RIP	Routing Information Protocol
SCTP	Stream Control Transmission Protocol
SGBD	Sistema Gestión de Base de Datos
sfacctd	Sflow accounting daemon
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VM	Virtual Machines
WAN	Wide Area Network

BIBLIOGRAFÍA

[1] Weathermap RedIRIS. Available from: <http://www.rediris.es/conectividad/weathermap/>

[2] Traffic Matrix, Abilene. Available from: <http://www.internet2.edu/network/>

[3]Traffic Matrix model, M.Roughan. Available from: <http://www.maths.adelaide.edu.au/matthew.roughan/traffic_matrices.html>

[4] Rincón,David; Raspall,Frederic; Sallent,Sebastià, Introduction to Network Traffic Analysis. Lecture notes, IOT course, EETAC-UCP.

[5] Net-SNMP, SNMP. Available from: <http://www.net-snmp.org/>

[6] Cisco NetFlow Collection Engine, NetFlow. Available from:<http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.html>

[7] Promiscuous Mode Accounting, Pmacct. Available from: <http://www.pmacct.net/>

[8] R. Hernando, 7 de Julio de 2002, Gestión de redes. Available from: <http://www2.rhernando.net/modules/tutorials/doc/redes/Gredes.html>

[9] Bastias López, Xavier, Evaluación de técnicas de captura y muestreo de tráfico en redes de ordenadores, TFC EETACT, 20-12-2012.

[10] NetFlow, Introducción a NetFlow. Available from <http://www.bibliociencias.cu/gsdl/collect/eventos/index/assoc/HASH010e.dir/doc.pdf>

[11] Networks. Configuring J-Flow Statistics. Available from: <http://www.juniper.net/techpubs/software/erx/junose82/swconfig-ipservices/html/ip-jflow-stats-config2.html.>

[12] Gray. GNU Cflow Manual. , 11-10-2011. Available from: <http://www.gnu.org/software/cflow/manual/index.html>

[13] Estébanez, Unai, Introducción a SNMP. Available from: <http://www.unainet.net/documents/SNMP.pdf>

[14] Lucente, Paolo. Available from: <paolo@pmacct.net>

[15] Libpcap, TCPDUMP & LIBPCAP. Available from: <http://www.tcpdump.org/>

[16] VirtualBOX, Oracle. Available from: <https://www.virtualbox.org/>

[17] Quagga, Quagga Routing Suite. Available from: <http://www.nongnu.org/quagga/>

[18] Fprobe, NetFlow probes. Available from: <http://fprobe.sourceforge.net/>

[19] MySQL, MySQL. Available from: <http://www.mysql.com/>

ANEXO I: INSTALACIÓN Y CONFIGURACIÓN DE LAS MÁQUINAS VIRTUALES

En este capítulo se va a describir como instalar el software de virtualización Oracle VM Virtualbox, como crear las máquinas virtuales y que configuración se ha realizado en cada una de ellas para poder interconetarlas.

Instalación VirtualBox

`VirtualBox` es un software que se encuentra en el repositorio de Ubuntu. Para instalarlo tenemos dos opciones que son:

a) Instalación desde el Centro de Software de Ubuntu (Figura 1)

b) Instalación desde la Shell con el comando: `apt-get install virtualbox.`

Fig.1 Instalación del software `VirtualBox` desde el Centro de Software de Ubuntu

Acceso a VirtualBox y creación de máquinas virtuales

En este apartado se va a explicar qué pasos se han de llevar a cabo para la creación de una máquina virtual.

Para acceder a `VirtualBox` abrimos un nuevo terminal `Ctrl+Alt+T` y desde la Shell con permisos de super usuario ejecutamos la aplicación (Figura 2):

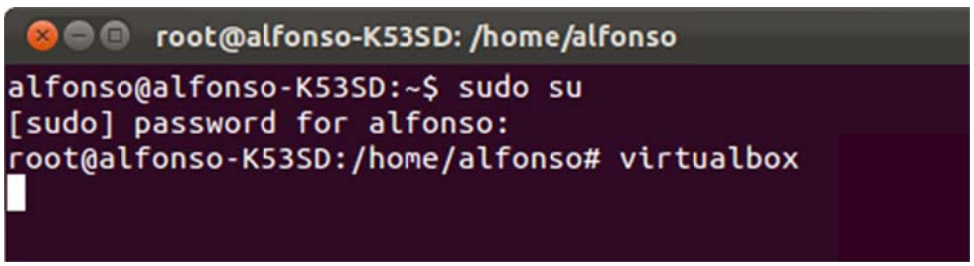


Fig.2 Consulta desde la shell para ejecutar software VirtualBox

Una vez realizada la consulta se abrirá la pantalla de administración de `Virtualbox` (Figura 3) dónde inicialmente solo estará habilitado el botón *Nueva* que nos permitirá crear una nueva VM.

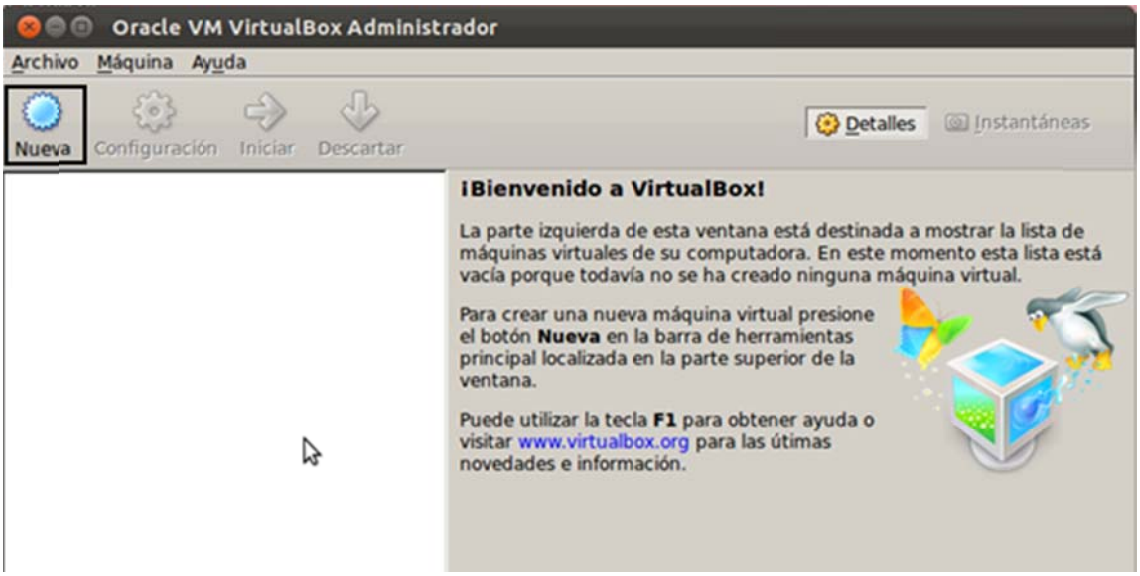


Fig. 3 Pantalla de administración del software `VirtualBox`

Pulsamos el botón *Nueva* y se abrirá el asistente para la creación de VM (Figura 4). Pulsamos el botón *Siguiente* y nos aparecerá la primera pantalla para crear nuestra VM, dónde tendremos que indicar el nombre de nuestra VM, el sistema operativo a utilizar y la versión de éste. En nuestro caso se ha elegido `SO Linux` y versión `Ubuntu`.

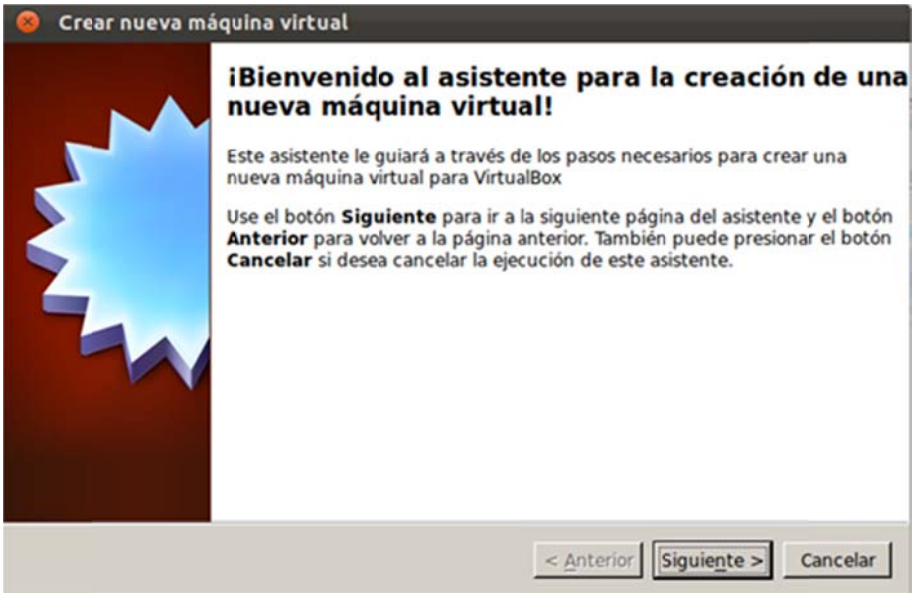


Fig.4 Asistente para la creación de una VM

Fig.5 Pantalla del asistente para asignar *Nombre* de la VM y versión del *SO*

Seguimos avanzando el asistente pulsando el botón *Siguiente* hasta llegar a la pantalla *Resumen* dónde se deberá pulsar el botón *Crear* para finalizar con la creación de una VM (Figura 6):

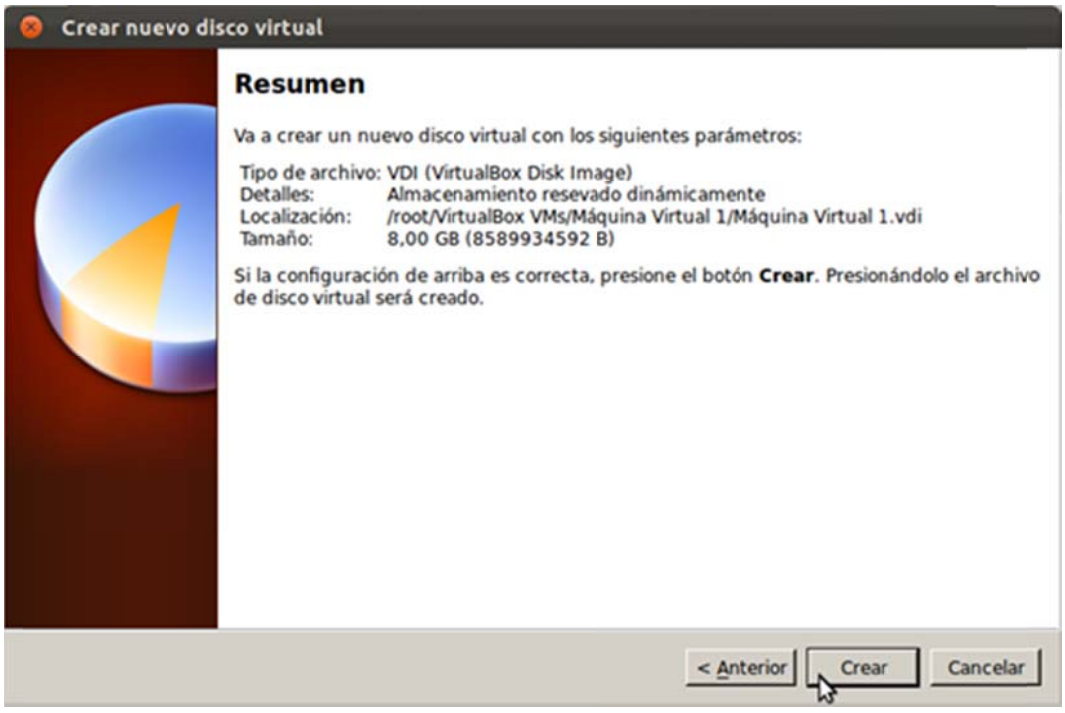


Fig.6 Pantalla final del asistente para crear VM con nombre Máquina Virtual 1

Una vez pulsado el botón *Crear*, volveremos a la pantalla de administración de `VirtualBox` dónde se podrá visualizar que se ha añadido la VM. Realizamos el mismo proceso hasta crear todas las VMs (Figura 7):

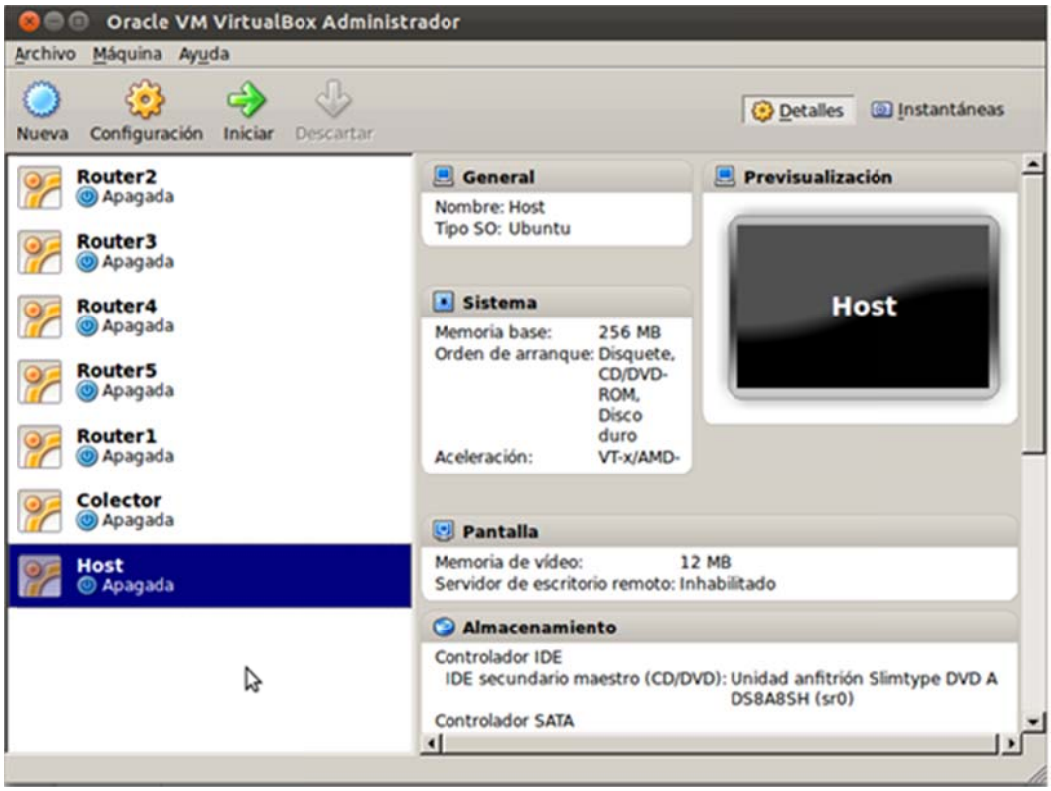


Fig.7 Pantalla administración de `VirtualBox` con todas las VMs ya creadas

Configuración máquinas virtuales para permitir interconexión

En este apartado vamos a detallar que pasos se han de seguir para configurar las VMs con el propósito de permitir la interconexión entre ellas. Cabe destacar que la mera configuración con el administrador de `VirtualBox` no va a permitir la interconexión entre máquinas. Ésta se va a encargar de simular un conector físico entre VMs.

El primer paso que se debe seguir es seleccionar la VM a configurar y pulsar el botón *Configuración*. Se abrirá la siguiente pantalla (Figura 8), dónde encontramos todos los parámetros que el software `VirtualBox` nos permite configurar. La propia aplicación, proporciona una descripción de que es cada campo.

`VirtualBox` permite dos tipos de interconexión: por un lado se permite la interconexión entre máquina anfitriona – máquina virtual y la segunda opción es la interconexión entre máquinas virtuales. Nos centramos en esta última.

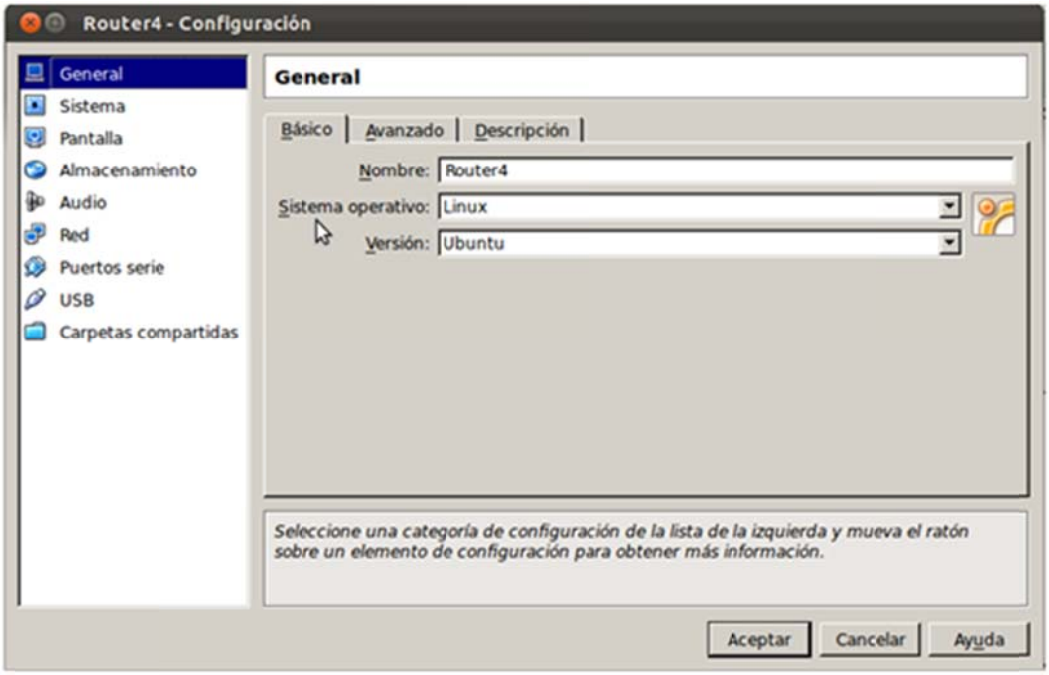


Fig.8 Pantalla de configuración de la VM

Vamos a enfocar la explicación en la sección *Red* ya que las demás secciones como *General, Sistema, Pantalla* entre otras se configuran a la hora de crear la VM y sus campos están descritos por el asistente. Accedemos a la sección *Red* y se nos abrirá la siguiente pantalla (Figura 9).

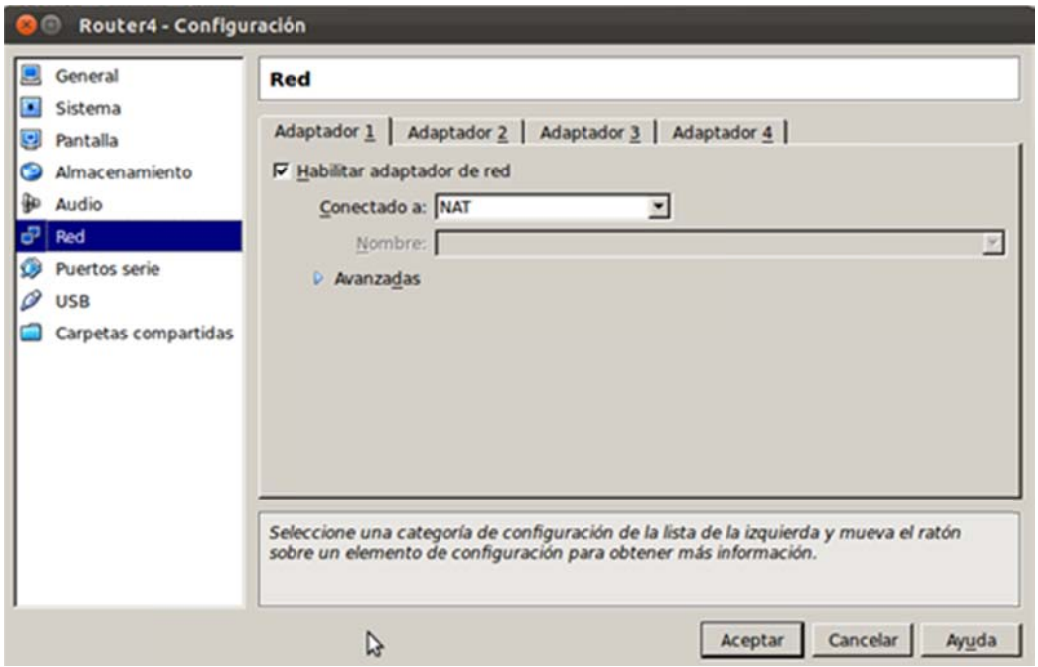


Fig.9 Pantalla de configuración de la sección *Red*

`VirtualBox` permite habilitar hasta 4 interfaces de red, presentadas como *Adaptador1, Adaptador2…* Para habilitar una interfaz de red, bastará con

seleccionar el adaptador que se quiere configurar y pulsar *Habilitar adaptador de red.*

`VirtualBox` presenta diferentes tipos de conexión:

- NAT → El servidor de `VirtualBox` asigna una dirección IP mediante DHCP para permitir la conectividad al exterior.
- Adaptador puente → Nos permitirá interconectar la máquina anfitriona con la máquina virtual. Se podrá configurar en tantas máquinas virtuales como interfaces físicas tenga la máquina.
- Red interna → Permite la interconexión entre VMs.

Nos centramos en éste último caso, ya que es el que se ha utilizado en este proyecto. El primer paso que se ha de realizar es habilitar el adaptador de red. El siguiente paso será desplegar el menú *Conectado a:* y seleccionar *Red interna.* Una vez realizado este cambio se habilitará el cajetín *Nombre.* En él deberemos introducir un nombre a modo de definición para posteriormente relacionarlo con la/s Vms que queremos interconectar.

La configuración final de cada VM ha sido:

- Máquina Colector (Figura 10):

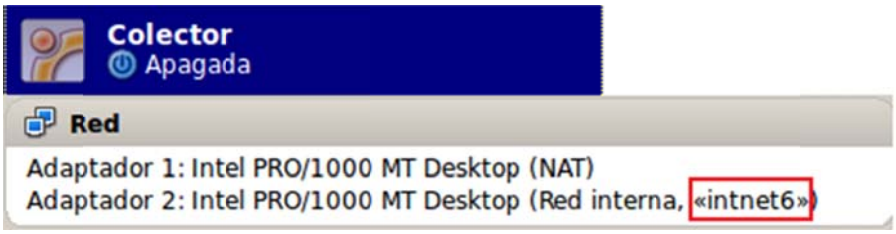


Fig.10 Configuración final de la sección Red de la VM Colector

- Máquina Router 5 (Figura 11):

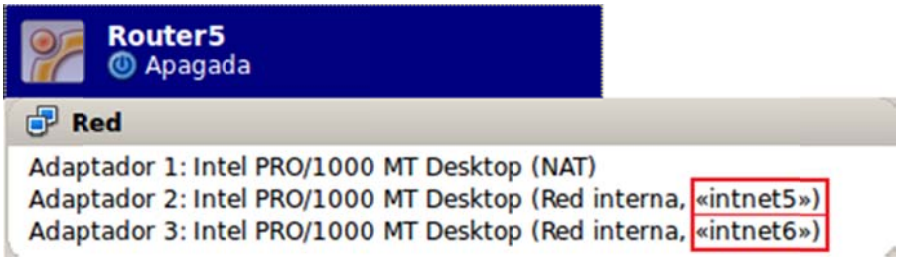


Fig.11 Configuración final de la sección Red de la VM Router5

- Máquina Router 1 (Figura 12):

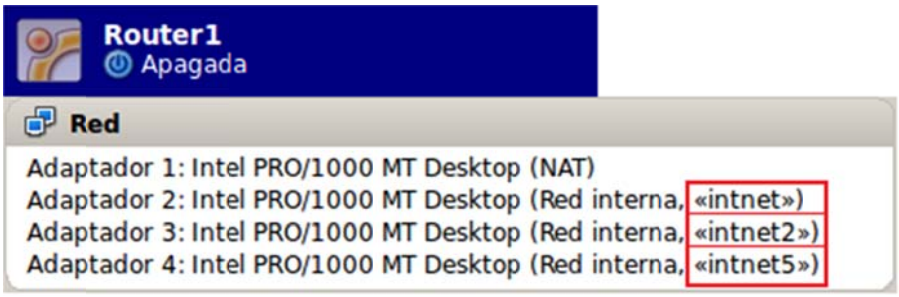


Fig.12 Configuración final de la sección Red de la VM Router1

- Máquina Router 2 (Figura 13):

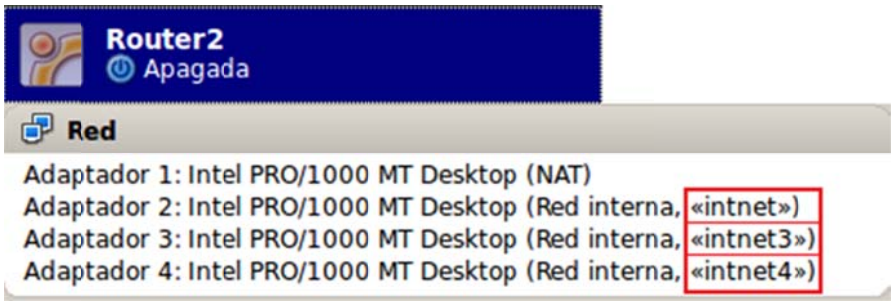


Fig.13 Configuración final de la sección Red de la VM Router2

- Máquina Router 3 (Figura 14):

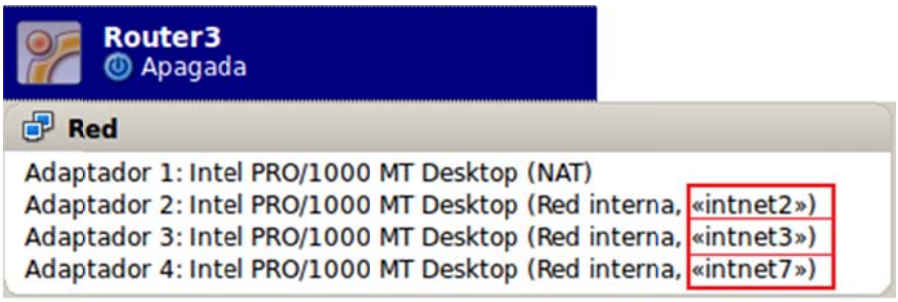


Fig.14 Configuración final de la sección Red de la VM Router3

- Máquina Router 4 (Figura 15):

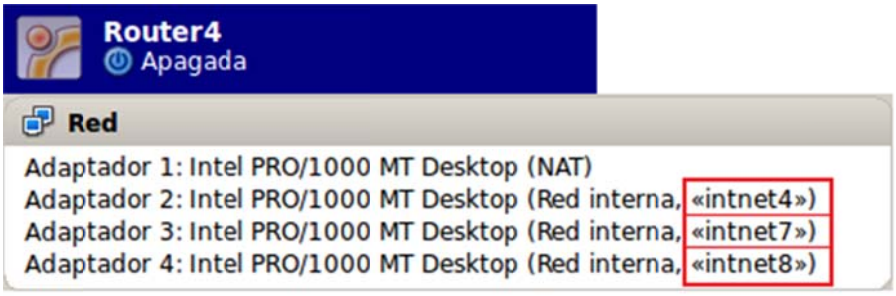


Fig.15 Configuración final de la sección Red de la VM Router4

- Máquina Host (Figura 16):

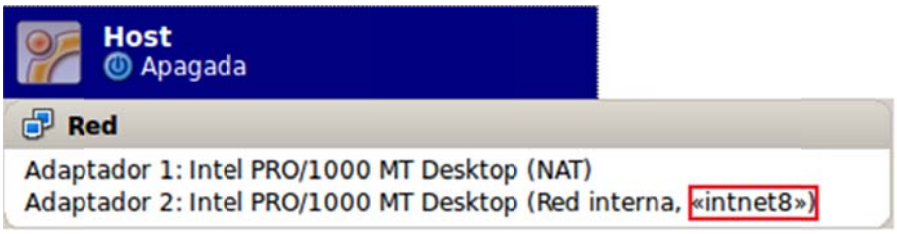


Fig.16 Configuración final de la sección Red de la VM Host

La interconexión final entre VMs quedará de la siguiente manera (Figura 17):

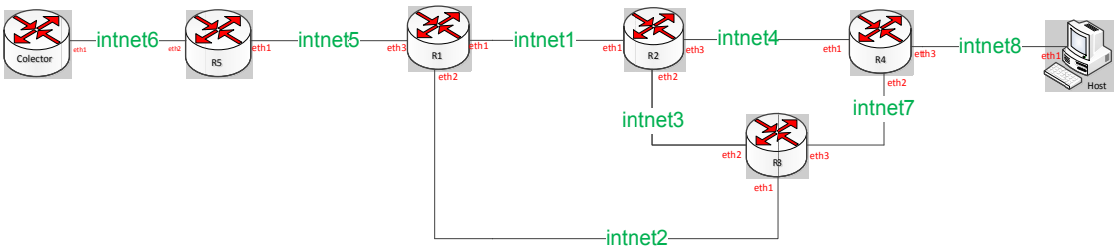


Fig.17 Interconexión física de las máquinas virtuales

ANEXO II: INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE quagga

En este capítulo se va a describir que pasos se han de seguir para instalar y configurar el software de encaminamiento quagga. Se presentarán los archivos de configuración creados en cada máquina virtual y sus tablas de encaminamiento.

Instalación del software quagga

Quagga es un software que se encuentra en el repositorio de Ubuntu. Como se describió en el capítulo anterior, se presentan dos maneras para realizar la instalación. Se realizará desde el Centro de Software de Ubuntu (Figura 18):

Fig.18 Instalación del software quagga desde el Centro de Software de Ubuntu

Una vez instalado abrimos un nuevo terminal “Ctrl+Alt+T” y accedemos al directorio donde se deben añadir los archivos de configuración. Quagga presenta unos ejemplos de configuración. Éstos se van a reutilizar y para ello los copiamos en el directorio donde se deben ejecutar los archivos de configuración. El siguiente paso que debemos realizar es cambiar el propietario y los permisos de los ficheros:

root@alfonso-K53SD:/home/alfonso# cd /etc/quagga/

#Copiamos ejemplos de configuración:

root@alfonso-K53SD:/etc/quagga# cp/usr/share/doc/quagga/examples/zebra.conf.sample zebra.conf

root@alfonso-K53SD:/etc/quagga# cp /usr/share/doc/quagga/examples/bgpd.conf.sample bgpd.conf

root@alfonso-K53SD:/etc/quagga# cp/usr/share/doc/quagga/examples/ospfd.conf.sample ospfd.conf

root@alfonso-K53SD:/etc/quagga# cp/usr/share/doc/quagga/examples/vtysh.conf.sample vtysh.conf

#Consultamos que se han copiado los paquetes:

root@alfonso-K53SD:/etc/quagga# **ls**

bgpd.conf daemons ospfd.conf zebra.conf debian.conf vtysh.conf

#Cambiamos el propietario y permisos a los ficheros:

root@alfonso-K53SD:/etc/quagga# chown quagga:quaggavty *.conf

root@alfonso-K53SD:/etc/quagga# chmod 640 *.conf

#Consultamos que se han realizado los cambios

root@alfonso-K53SD:/etc/quagga# ls -l

-rw-r----- 1 quagga quaggavty 582 2012-12-17 15:33 bgpd.conf

-rw-r----- 1 quagga quaggavty 583 2012-10-01 16:14 bgpd.conf.sav

```
-rw-r--r-- 1 root   root      850 2012-10-01 16:13 daemons

-rw-r----- 1 quagga quaggavty 471 2012-05-05 23:23 debian.conf

-rw-r----- 1 quagga quaggavty 182 2012-12-17 15:33 ospfd.conf

-rw-r----- 1 quagga quaggavty 126 2012-12-17 15:33 vtysh.conf

-rw-r----- 1 quagga quaggavty 385 2012-12-17 15:33 zebra.conf
```

Quagga presenta una serie de *daemons*, tanto para encaminamiento interno como encaminamiento externo. Para activarlos hay que acceder al archivo daemons.conf que se encuentra en el directorio /etc/quagga. A continuación se va a mostrar cual es el contenido del fichero de configuración y que cambios hay que realizar para activar un *daemon.*

```
root@alfonso-K53SD:/etc/quagga# nano daemons

#

# ATTENTION:

#

# When activation a daemon at the first time, a config file, even if it is

# empty, has to be present *and* be owned by the user and group "quagga", else

# the daemon will not be started by /etc/init.d/quagga. The permissions should

# be u=rw,g=r,o=.

# When using "vtysh" such a config file is also needed. It should be owned by

# group "quaggavty" and set to ug=rw,o= though. Check /etc/pam.d/quagga, too.

#

zebra=yes

bgpd=yes
```

```
ospfd=no

ospf6d=no

ripd=no

ripngd=no

isisd=no

#En este caso hemos activado los daemons zebra y bgpd
```

Una vez realizado estos cambios, ya se puede pasar a modificar los archivos de configuración de cada *daemon* en cada VM. A continuación vamos a presentar todos los archivos de configuración de cada VM y para finalizar presentaremos sus tablas de encaminamiento:

- Máquina Colector (Figura 19):

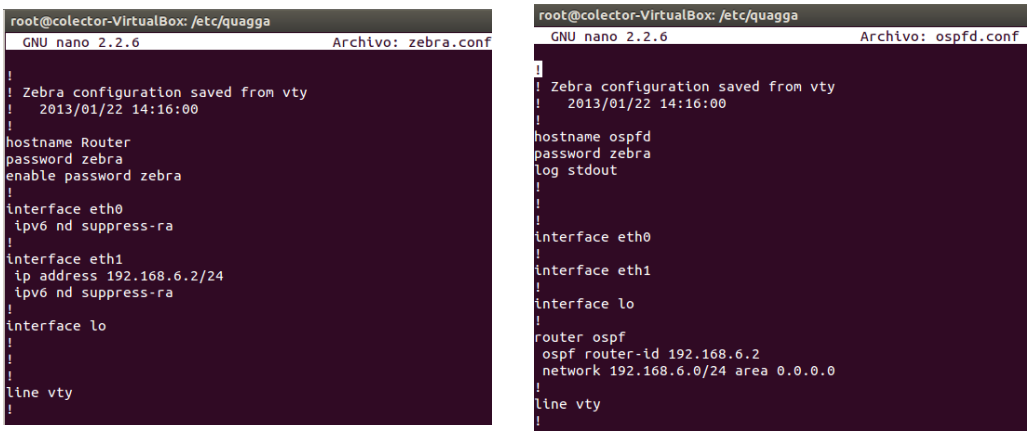


Fig.19 Archivos de conf. en la máquina Colector: zebra – ospf

- Máquina R5 (Figura 20):

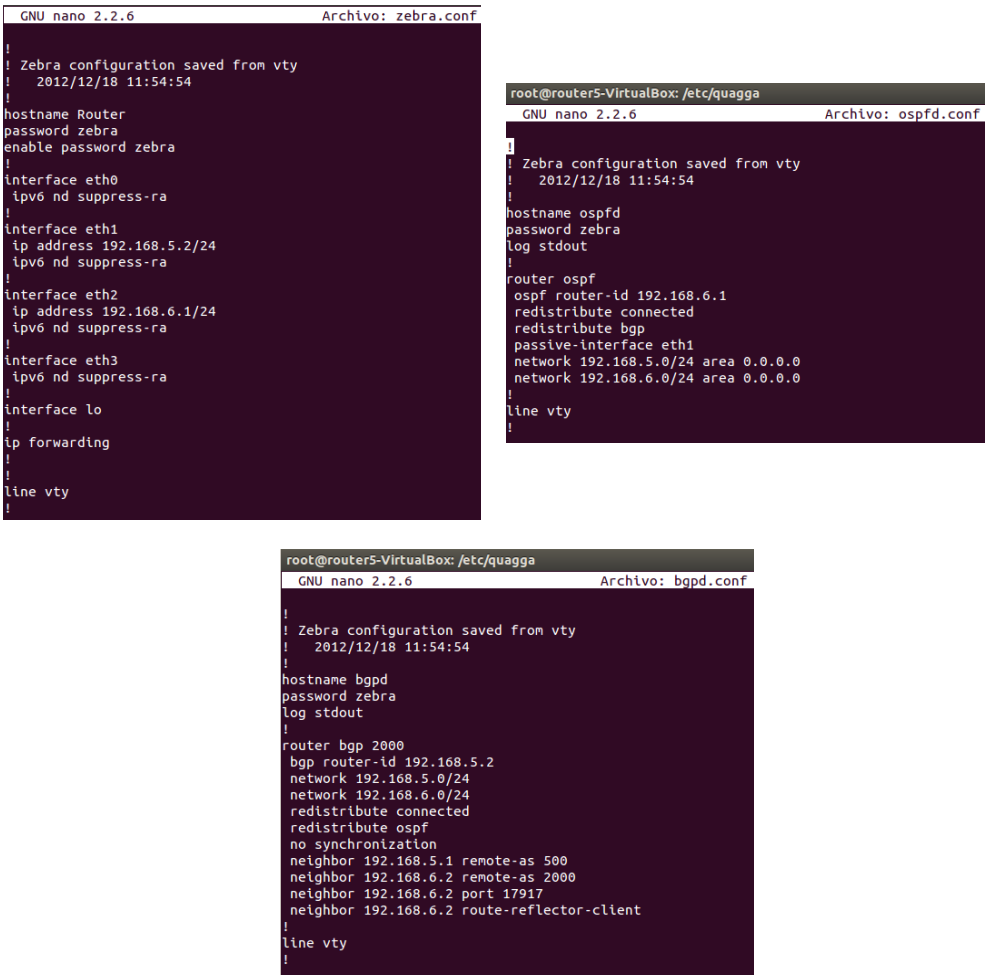


Fig.20 Archivos de conf. en la máquina R5: zebra - ospf - bgp

- Máquina R1 (Figura 21):

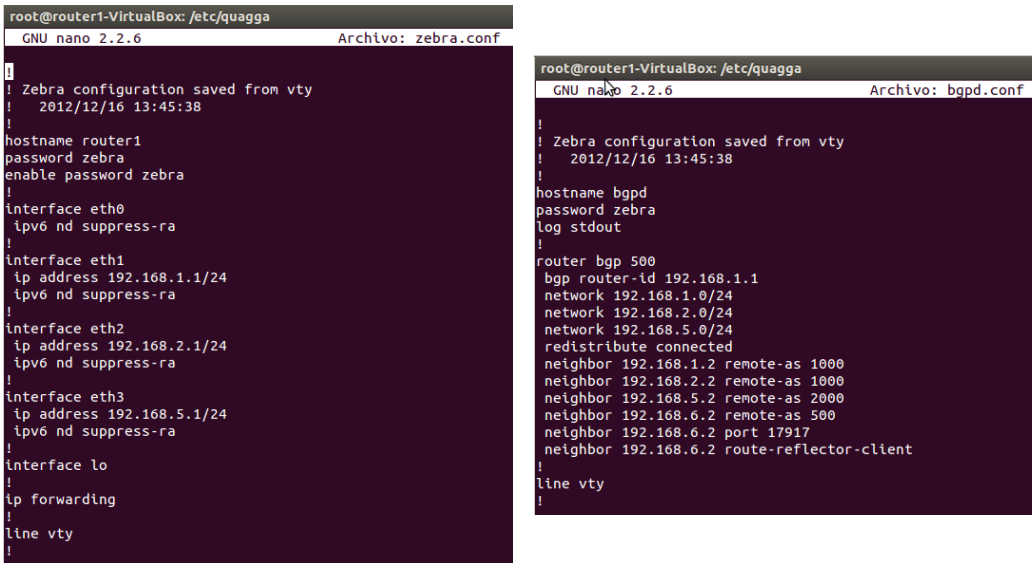


Fig.21 Archivos de conf. en la máquina R1: zebra – bgp

- Máquina R2 (Figura 22):

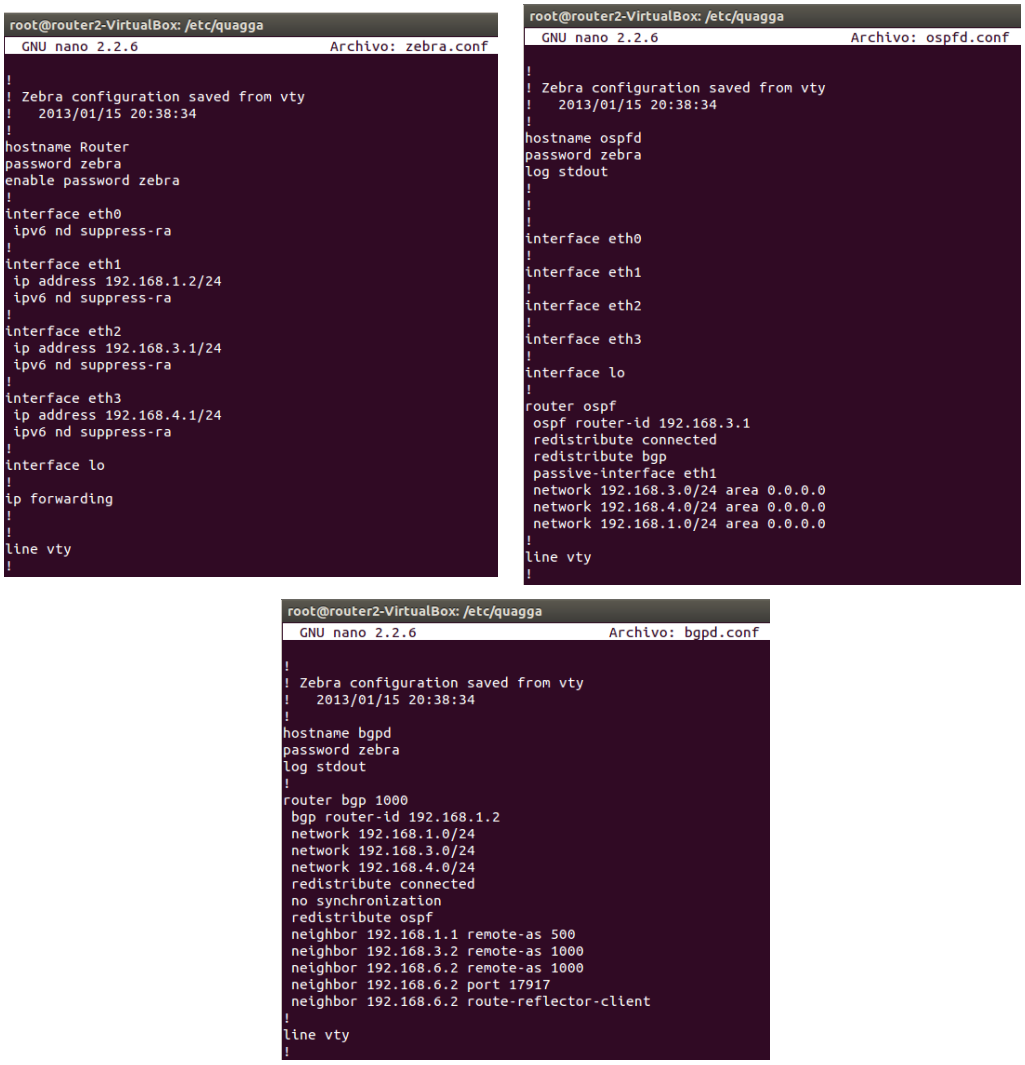


Fig.22 Archivos de conf. en la máquina R2: zebra - ospf – bgp

- Máquina R3 (Figura 23):

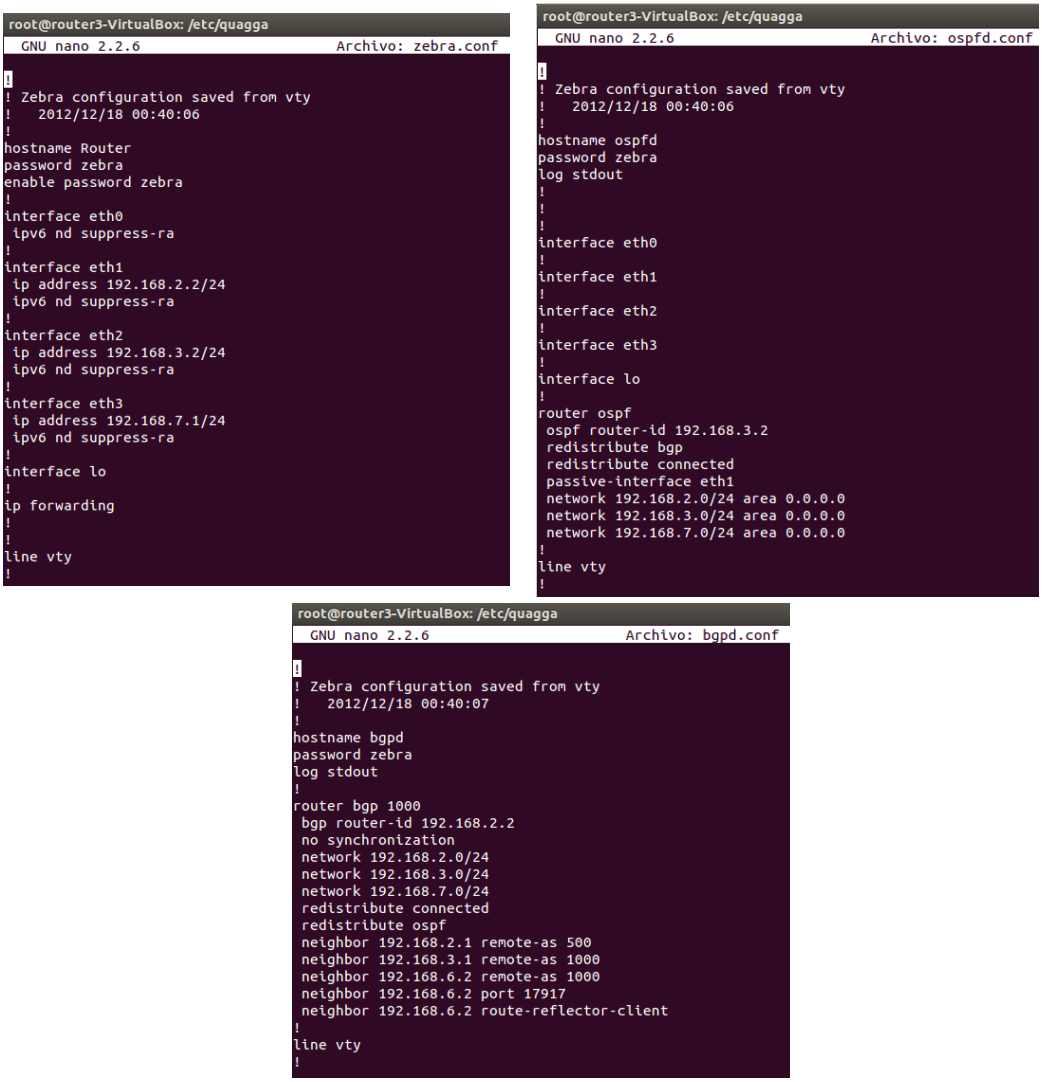


Fig.23 Archivos de conf. en la máquina R3: zebra - ospf – bgp

- Máquina R4 (Figura 24):

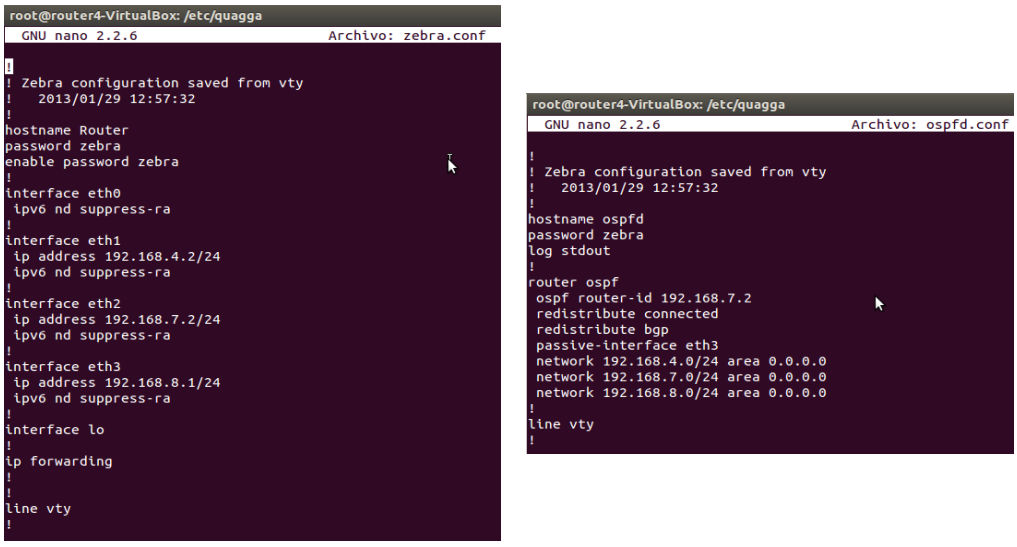


Fig.24 Archivos de conf. en la máquina R4: zebra – ospf

A continuación mostramos la tabla de encaminamiento de cada una de las VM una vez ejecutados los *daemons:*

- Máquina Colector (Figura 25):

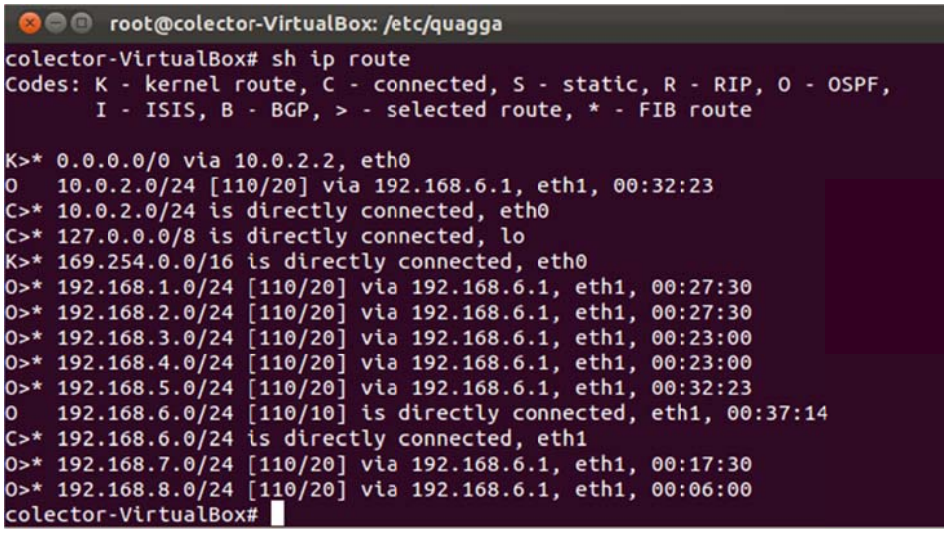


Fig.25 Tabla de encaminamiento de la máquina Colector

- Máquina R5 (Figura 26):

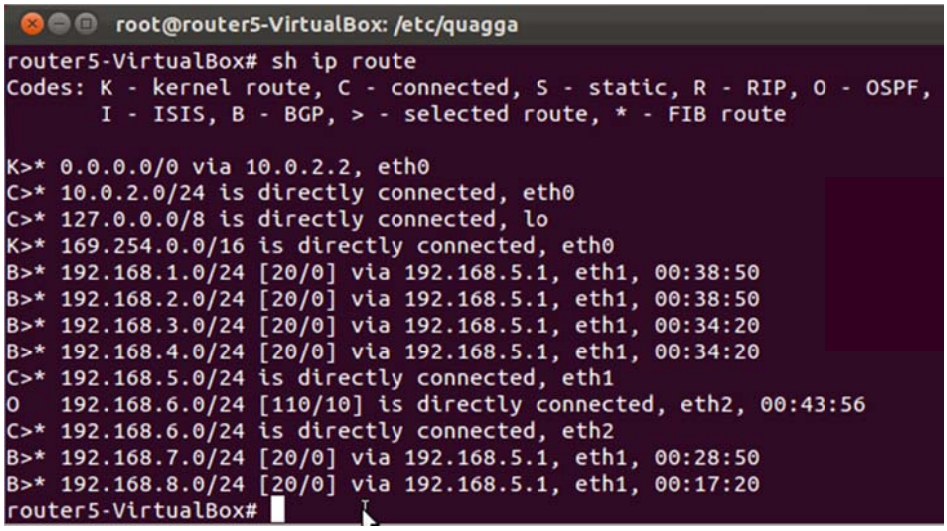


Fig.26. Tabla de encaminamiento de la máquina R4

- Máquina R1 (Figura 27):

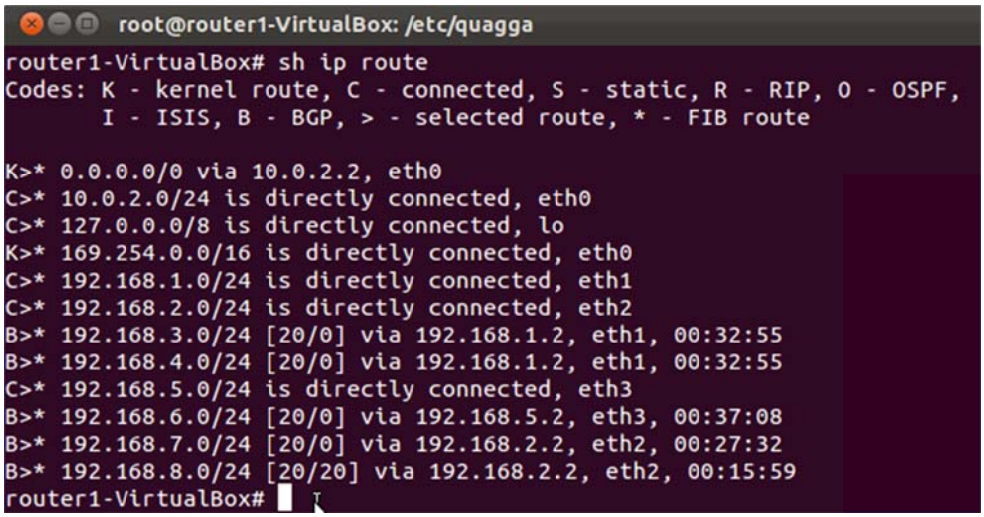


Fig.27 Tabla de encaminamiento de la máquina R1

- Máquina R2 (Figura 28):

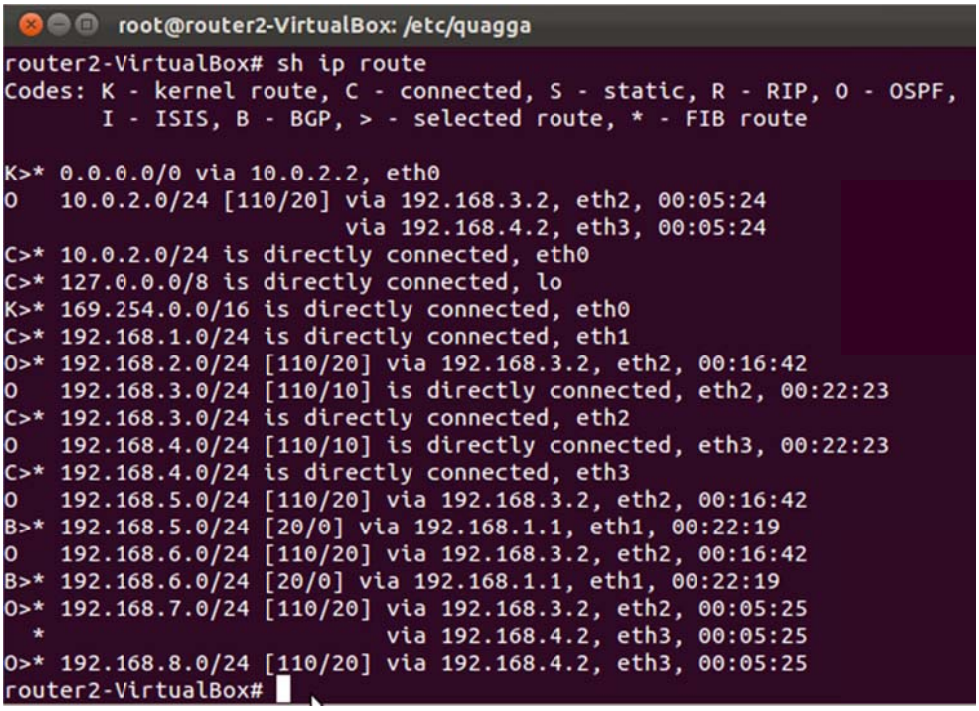


Fig.28 Tabla de encaminamiento de la máquina R2

- Máquina R3 (Figura 29):

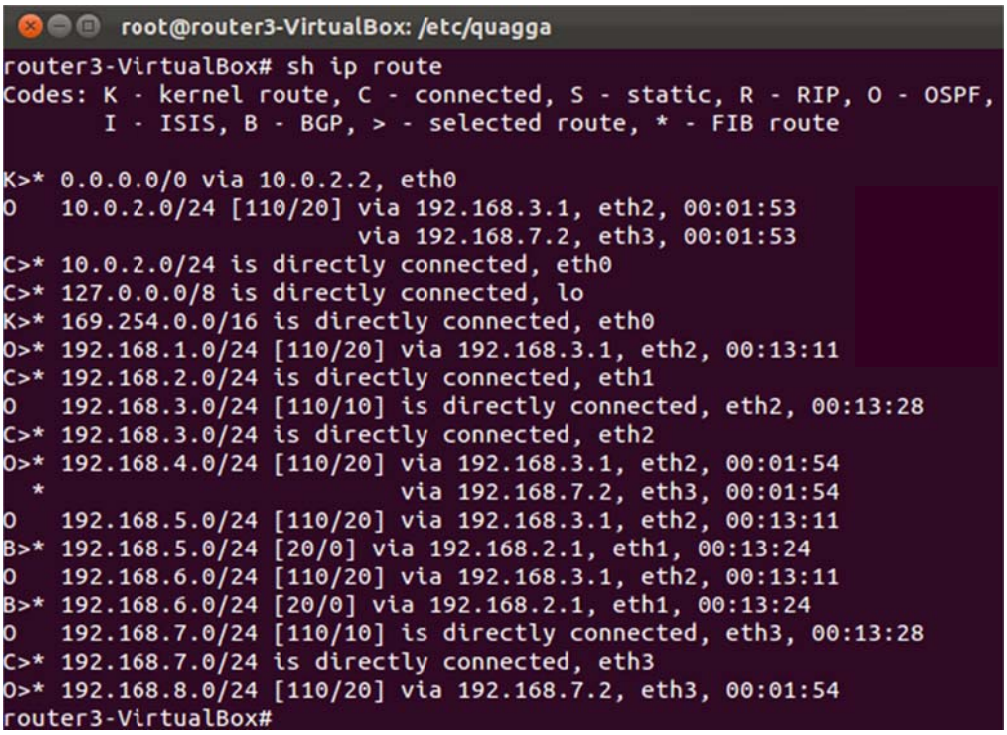


Fig.29 Tabla de encaminamiento de la máquina R3

- Máquina R4 (Figura 30):

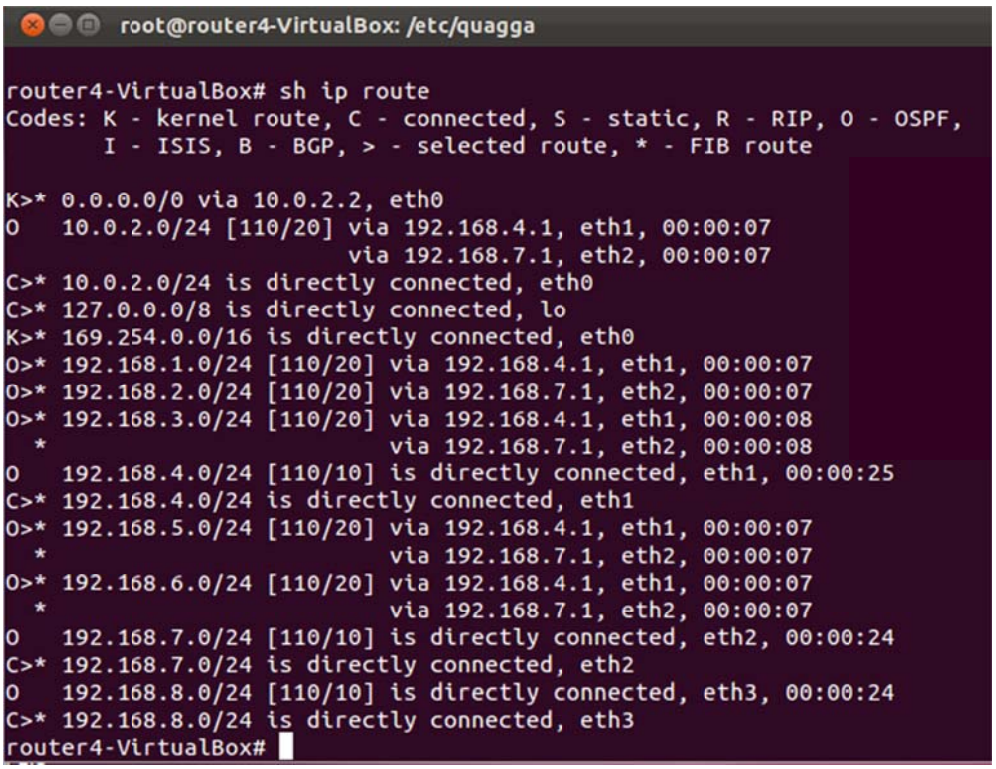


Fig.30 Tabla de encaminamiento de la máquina R4

ANEXO III: INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE pmacct

En esta capítulo vamos a presentar que pasos se han seguido a la hora de instalar el software `pmacct` y que archivos de configuración se han creado en las VMs dónde se va a ejecutar.

Instalación del software pmacct

La versión del software `pmacct` que se presenta en el repositorio de Ubuntu, es una versión antigua y aconsejado por el diseñador del software se aconsejo instalar la última versión que la podemos encontrar en la web corporativa de pmacct http://www.pmacct.net.

El primer paso que debemos realizar es acceder a la sección *Requeriments* e instalar los requisitos que `pmacct` necesita para completar su instalación. Los paquetes que se deben instalar son:

- tcpdump-4.3.0.tar.gz
- libpcap-1.3.0.tar.gz

Descargamos los paquetes y procedemos a su instalación desde la shell siguiendo estos pasos:

1) Descomprimir el paquete: root@router1-VirtualBox:/home/router1 /Descargas# `tar xvfz libpcap-1.3.0.tar.gz`

2) Accedemos al paquete descomprimido: root@router1-VirtualBox:/home /router1/Descargas# `cd libpcap-1.3.0`

3) Compilamos el paquete para preparar la instalación: root@router1-VirtualBox:/home/router1/Descargas/libpcap-1.3.0# `./configure`

4) Instalamos el paquete: root@router1-VirtualBox:/home/router1/ Descargas/libpcap-1.3.0# `make` ***//*** root@router1-VirtualBox:/home/ router1/Descargas/libpcap-1.3.0# `make install`

Realizamos estos mismos pasos para instalar el paquete tcpdump-4.3.0.tar.gz.

Una vez instaladas todas las dependencias de `pmacct`, procedemos a instalar el paquete. Descargamos la última versión vigente que es *pmacct-0.14.2.tar.gz.* Los pasos para realizar la instalación son los mismos comentados anteriormente:

1) Descomprimimos el paquete: root@router1-VirtualBox:/home/router1 /Descargas# `tar xvfz pmacct-0.14.2.tar.gz`

2) Accedemos al paquete descomprimido: root@router1-VirtualBox:/ home /router1/Descargas# `cd pmacct-0.14.2`

3) Compilamos el paquete para preparar la instalación: root@router1-VirtualBox:/home/router1/Descargas/pmacct-0.14.2#`./configure –enable-threads –enable-mysql` Se habilita multi hilos para poder ejecutar el daemon BGP y mysql para poder utilizar la base de datos mysql.

4) Instalamos el paquete :root@router1VirtualBox:/home/router1/Descargas/pmacct-0.14.2# `make` *//* root@router1-VirtualBox:/home/router1/Descargas/pmacct-0.14.2# `make install`

Una vez instalado el software `pmacct` en las VMs Colector, R5, R1 y R2 pasamos a describir los archivos de configuración que se han creado.

- Archivo de configuración de *daemon* NetFlow en la máquina Colector (Figura 31):

```
  GNU nano 2.2.6                Archivo: nfacctd1.conf                Modificado

!
! nfacctd configuration example
!
! Did you know CONFIG-KEYS contains the detailed list of all configuration keys
! supported by 'nfacctd' and 'pmacctd' ?
!
! debug: true
daemonize: false

!PRIMITIVAS --> SE VAN A RECOGER LOS SIGUIENTES DATOS!
aggregate: src_host, dst_host, src_as, dst_as, src_port, dst_port, as_path, peer_src_ip, peer_dst_ip,
peer_src_as, peer_dst_as, flows

!PLUGINS A ACTIVAR --> se activa plugin mysql, ya que es la DB donde vamos a guardar los registros recolectados
plugins: mysql

!CONFIGURACIÓN DB MYSQL!
sql_db: pmacct
sql_table: acct_bgp2
sql_history: 5m
sql_refresh_time: 300
sql_table_version: 4
sql_table_type: bgp
sql_host: 127.0.0.1
sql_user: root
sql_passwd: bluesky88
sql_history_roundoff: m
sql_optimize_clauses: true

!CONFIGURACIÓN COLECTOR NETFLOW!

nfacctd_port: 2021
nfacctd_ip: 192.168.6.2
nfacctd_as_new: bgp
nfacctd_net: bgp
nfacctd_time_new: true

!CONFIGURACIÓN DAEMON BGP!

bgp_daemon: true
bgp_daemon_ip: 192.168.6.2
bgp_daemon_port: 17917
bgp_daemon_max_peers: 6
bgp_daemon_msglog: true
bgp_peer_src_as_type: bgp
bgp_src_as_path_type: bgp
```

Fig.31 Archivo de configuración daemon `nfacctd` en la máquina Colector

- Archivo de configuración del agente NetFlow en la máquina R5 (Figura 32):

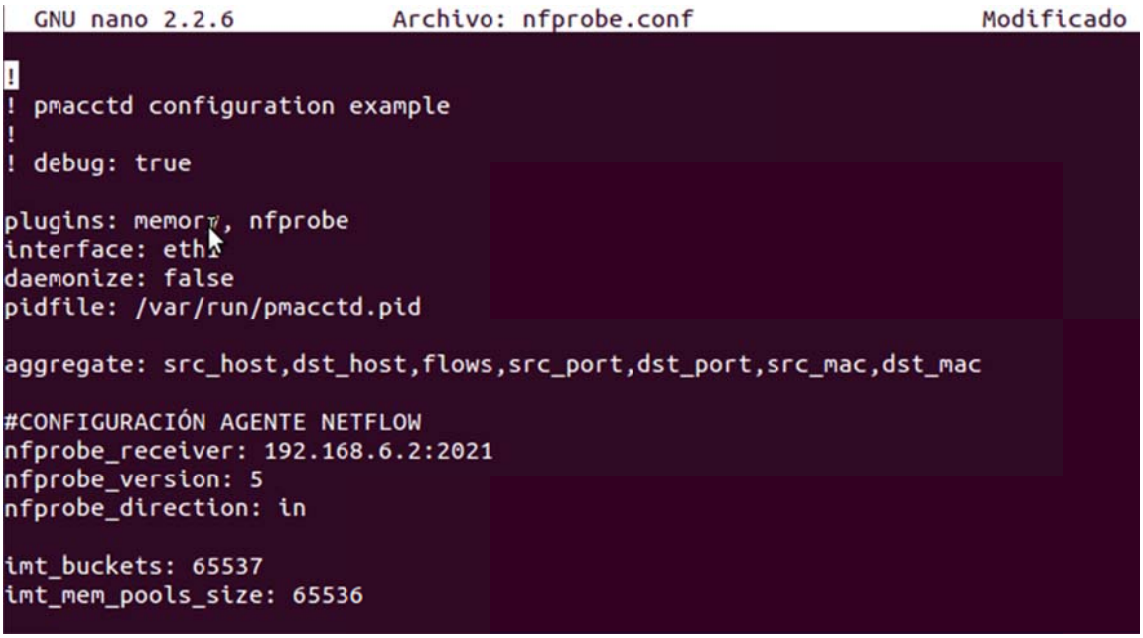


Fig.32 Archivo de configuración del agente NetFlow en R5

- Archivo de configuración del daemon `nfacctd` en la máquina R1 (Figura 33).

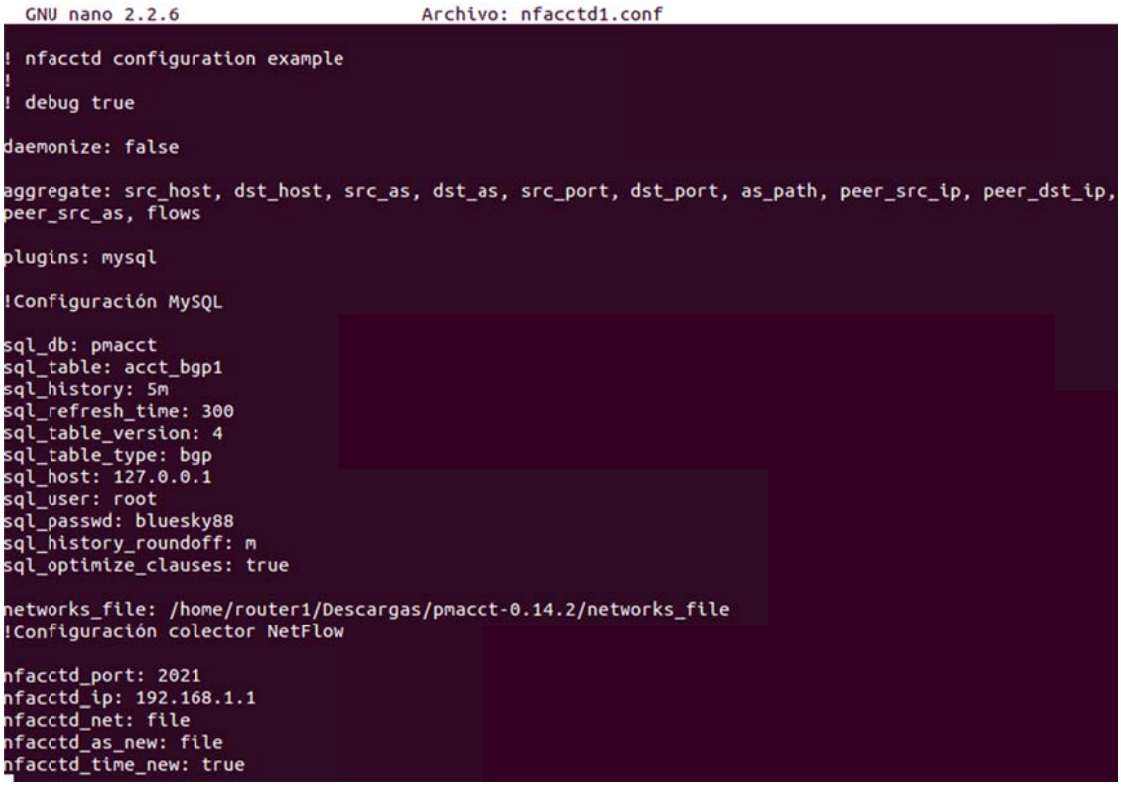


Fig.33 Archivo de configuración del daemon `nfacctd` en la máquina R1

- Archivo de configuración del daemon `nfacctd` en la máquina R2 (Figura 34).

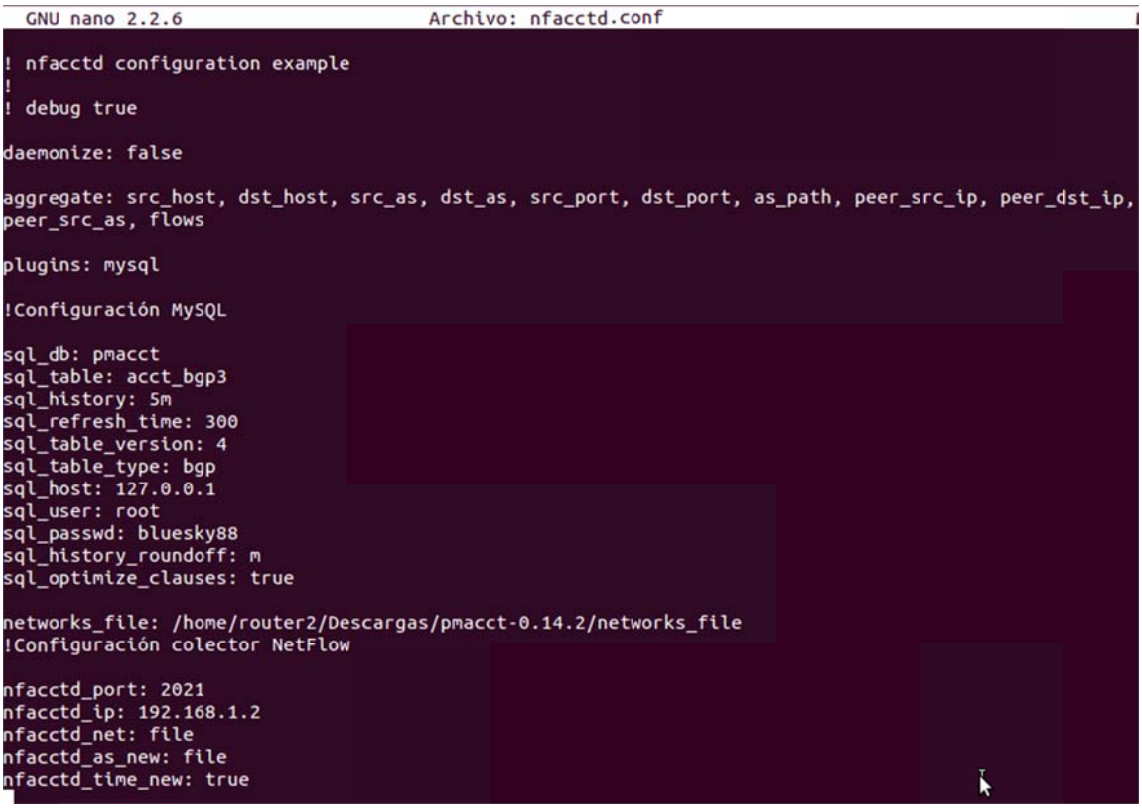


Fig.34 Archivo de configuración del daemon `nfacctd` en la máquina R2